

Resource analysis for quantum-aided Byzantine agreement with the four-qubit singlet state

Zoltán Guba¹, István Finta^{2,3}, Ákos Budai^{1,4,2}, Lóránt Farkas², Zoltán Zimborás^{4,5}, and András Pályi¹

¹Department of Theoretical Physics, Institute of Physics, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary

²Nokia Bell Labs

³Óbuda University

⁴Wigner Research Centre for Physics, H-1525 Budapest, P.O.Box 49, Hungary

⁵Eötvös University, Budapest, Hungary

In distributed computing, a Byzantine fault is a condition where a component behaves inconsistently, showing different symptoms to different components of the system. Consensus among the correct components can be reached by appropriately crafted communication protocols even in the presence of byzantine faults. Quantum-aided protocols built upon distributed entangled quantum states are worth considering, as they are more resilient than traditional ones. Based on earlier ideas, here we establish a parameter-dependent family of quantum-aided weak broadcast protocols. We compute upper bounds on the failure probability of the protocol, and define and illustrate a procedure that minimizes the quantum resource requirements. Following earlier work demonstrating the suitability of noisy intermediate scale quantum (NISQ) devices for the study of quantum networks, we experimentally create our resource quantum state on publicly available quantum computers. Our work highlights important engineering aspects of the future deployment of quantum communication protocols with multi-qubit entangled states.

Contents

1	Introduction	1
2	A two-parameter Weak Broadcast protocol	2
3	Security of Weak Broadcast	4
4	Resource optimization of the Weak Broadcast protocol	5
4.1	Tight upper bound for the failure probability – analytical results	5
4.2	Tight upper bound for the failure probability - Monte-Carlo simulations	5

4.3	Optimization in the parameter space	6
5	Four-qubit singlet state on IBM Q and IonQ devices	7
5.1	Circuit A	7
5.2	Circuit B	8
6	Conclusion and outlook	9
	Acknowledgments	9
	References	9
A	Broadcast, weak broadcast, and their truth tables	11
B	Adversary strategies against the Weak Broadcast protocol	11
B.1	S faulty	12
B.2	R_0 faulty	15
C	Failure probabilities	17
C.1	No faulty	18
C.2	S faulty	18
C.3	R_0 faulty	19
D	Proofs of optimality	20
D.1	S faulty	20
D.2	R_0 faulty	21
E	Security proof	21
E.1	No faulty	21
E.2	S faulty	22
E.3	R_0 faulty	23
F	Preparation and benchmarking of the four-qubit singlet state on IBM Q	24

1 Introduction

In distributed computing, a Byzantine fault is a condition where a component behaves inconsistently, show-

ing different symptoms to different components of the system. To reach consensus among the correctly functioning components, communication protocols resilient to such faults must be used.

An early example of such protocols was developed by Pease et al. [1]. That protocol provides consensus if $t < n/3$, where n is the number of components in the distributed system, and t is the maximum number of components exhibiting Byzantine fault. For example, if the maximum number of faulty components is 1, then at least 4 components are needed to reach consensus. We will refer to this property of the protocol as *resilience*. The efficiency of early protocols [1, 2] has since been improved [3–6].

Quantum-aided ‘weak broadcast’ protocols developed in the past two decades, built upon distributed entangled quantum states [7] such as the four-qubit singlet state [8, 9], are worth considering, as they offer a higher resilience: t faulty components are tolerated as long as $t < n/2$. For example, to be resilient against $t = 1$ faulty component, having $n = 3$ components is sufficient.

In this work, we build on those earlier ideas, and introduce a parameter-dependent family of quantum-aided weak broadcast protocols that relies on an entangled four-qubit singlet state [8]. Our main contributions are as follows.

(1) *Proof*. We prove the weak broadcast functionality of the protocol in a certain range of protocol parameters. The proof relies on considering the failure probability $p_f(m)$ as the function of the number m of resource states, and upper bounding this failure probability exponentially, that is, proving $p_f \leq e^{-bm}$ with a certain $b > 0$ as $m \rightarrow \infty$.

(2) *Tight upper bound for failure probability*. We compute a tight upper bound to the failure probability of the protocol, as function of the number of quantum states used to broadcast a single classical bit.

(3) *Resource optimisation*. We use the tight upper bound to perform a resource optimisation, i.e., compute the answer to these questions: (i) For a given allowed failure probability, how to set the protocol parameters to minimise the number of resource states? (ii) What is the number of required resources states?

(4) *Preparation of the four-qubit singlet state*. Following earlier work demonstrating the suitability of noisy intermediate-scale quantum (NISQ) devices for the study of quantum networks [10], we experimentally create our resource quantum state on the public quantum-computer prototypes of IBM and IonQ, and characterise its state fidelity.

In a spirit similar to a recent study [11] of an alternative quantum-aided protocol [12], our analysis illustrates multiple engineering aspects of future deployment of such protocols in practice.

The rest of the paper is structured as follows. In Sec. 2, we define a two-parameter weak broadcast protocol based on the four-qubit singlet state. In Sec. 3,

we prove the functionality of the protocol. In Sec. 4, we outline how to compute the tight upper bound for the failure probability, and discuss the resource optimisation protocol. In Sec. 5, we construct two four-qubit quantum circuits that can be used to generate the singlet state, and implement those circuits on IBM’s quantum computer devices. We discuss our results and follow-up ideas, and draw conclusions, in Sec. 6.

2 A two-parameter Weak Broadcast protocol

Multi-party communication protocols exist for various functionalities. Two examples are broadcast and weak broadcast [13]. An important building block for more complex protocols is a weak broadcast protocol that is resilient up to one faulty component in the presence of 3 components. Following Ref. [13], we will denote such protocols as WBC(3,1). A key result of Ref. [13] is that WBC(3,1) can be used as a primitive, to efficiently build a broadcast protocol with $n \geq 3$ components, resilient against any $t < n/2$ faulty components. This motivates us to define and study a protocol achieving WBC(3,1).

In a distributed system with 3 components, the WBC(3,1) protocol has the following goal: the sender (S) wants to distribute a single bit $x_S \in \{0, 1\}$ of classical information to the two receivers R_0 and R_1 , in such a way that there is a consensus about the bit value even if one of the three components is an active adversary. More formally, in a WBC(3,1) protocol, the output of the receivers can take a value \perp (often called ‘abort’) besides 0 and 1, and the protocol should fulfill the conditions of ‘Validity’ and ‘Consistency’. Validity means that if the Sender S is correct (i.e., exactly follows the prescribed protocol), then the output bit values of all correct components are equal to x_S . Consistency means that if any correct component has an output $y \in \{0, 1\}$, then all other correct components should have an output from $\{y, \perp\}$. (See a more detailed description in Appendix A.)

The two-parameter WBC(3,1) protocol we define here is derived from similar protocols in earlier works [7–9]. It prescribes communication between three components (S, R_0, R_1) to achieve consensus in the sense of WBC(3,1) described above. The quantum resource of this protocol is a four-qubit singlet state [8]:

$$|\psi\rangle = \frac{1}{2\sqrt{3}}(2|0011\rangle - |0101\rangle - |0110\rangle - |1010\rangle - |1001\rangle + 2|1100\rangle). \quad (1)$$

In fact, we assume that to achieve WBC(3,1) with a single data bit, the parties have distributed $m \in \mathbb{Z}^+$ four-qubit singlet states through public quantum channels such that the first two qubits are distributed

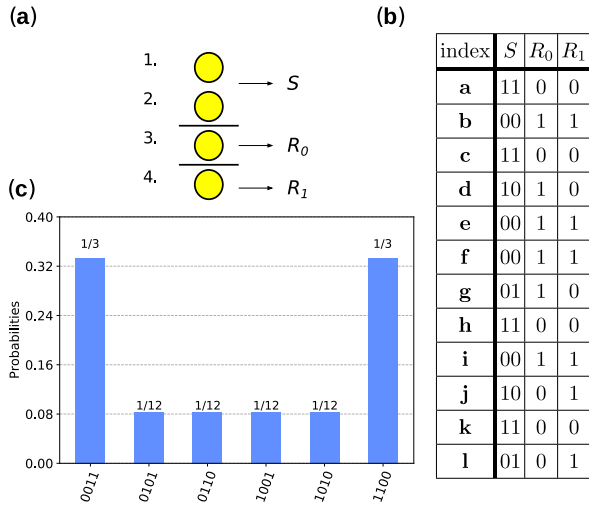


Figure 1: Using a four-qubit singlet state for weak broadcast. (a) Yellow circles indicate the four qubits of the singlet state $|\psi\rangle$ in Eq. (1). In the protocol, the qubits are distributed: qubits 1 and 2 are with the Sender (S), qubit 3 is with R_0 , and qubit 4 is with R_1 . (b) Example of measurement outcomes from $m = 12$ four-qubit singlet states. Such a table of data is called an Event. (c) Probability density function of the six possible outcomes when the four-qubit singlet state $|\psi\rangle$ is measured.

to S , the third qubit is distributed to R_0 , and the last qubit is distributed to R_1 , as shown in Fig. 1a. In rough terms, the more four-qubit singlet states are used to broadcast one bit, the higher the probability of success; hence the integer m is a key quantity in assessing the performance of the protocol.

Having m singlet states distributed, the components perform measurements in the computational basis, each obtaining its own random list of bit values: S obtains m bit-pairs, while R_0 and R_1 obtain m bits each. An example for such a collection of random correlated bitstrings for $m = 12$ is shown in Fig. 1b, and the probability density function of the six possible outcomes of four-bit strings, derived straightforwardly from Eq. (1), is shown in Fig. 1c. We will refer to the six possible four-long bitstring outcomes as *Elementary Events*, and we will refer to a particular instance of this collection of random data, as shown in Fig. 1b, as an *Event*. In what follows, we will use $Q_j[\alpha]$ to denote the measurement result of component j in its row α , where $j \in (S, R_0, R_1) \equiv (S, 0, 1)$ and $\alpha \in (1, 2, \dots, m) \equiv (a, b, \dots)$. For example, in Fig. 1c, we see $Q_S[d] = 10$, $Q_0[e] = 1$, etc.

The Weak Broadcast protocol presented below has two real-valued parameters: $0 < \mu < 1/3$ and $1/2 < \lambda < 1$. We anticipate that the protocol works well if μ and λ are close to their upper bound.

The Weak Broadcast protocol we propose is as follows. In all steps, the classical channels are assumed to be authenticated, and all the channels, the source of quantum states, and the measurements, are assumed

to be perfect.

- Invocation Phase.** S sends the data bit (x_S) to be broadcasted to the other components. We denote the bits of the components R_0 and R_1 received as $x_0 = x_S$ and $x_1 = x_S$. Now the Sender measures all its qubits (2 qubits in each distributed state) in the computational basis. For each qubit pair of S , if both measurements corresponding to state index α (see Fig. 1b) yielded x_S , then S adds index α to its *check set* σ_S . After assembling σ_S , this check set is sent to both receivers. At this point, both receivers hold a bit value x_j and a set σ_j from the Sender. The Sender also sets its output to $y_S = x_S$. In the example of Fig. 1b, and assuming $x_S = 0$, the check set is $\sigma_S = \{b, e, f, i\}$.
- Check Phase.** Now both R_0 and R_1 check the consistency of their data received from S . For this, R_j measures all of its qubits in the check set σ_j , and if all the results differ from x_j (*Consistency Condition*), and also the check set is large enough (*Length Condition*), then it accepts the message x_j . For R_0 , this implies that it chooses its output to be $y_0 = x_0$. For R_1 , it chooses the value of an intermediate variable $\tilde{y}_1 = x_1$. The check set is large enough if the number of its elements is at least $T \equiv \lceil \mu \cdot m \rceil$, where $0 < \mu < 1/3$. If R_0 finds that any of the two conditions is violated, then it sets its output to ‘abort’, $y_0 = \perp$. If R_1 finds that any of the two conditions is violated, then it sets its intermediate value to ‘abort’, $\tilde{y}_1 = \perp$.
- Cross-calling Phase.** R_0 sends to R_1 its output value y_0 and the check set σ_0 it received from S . R_1 receives these as y_{01} and ρ_{01} , respectively.
- Cross-check Phase.** R_1 evaluates the following three conditions and if all of them are true, then it outputs the value received from R_0 , that is, $y_1 = y_{01}$; otherwise it outputs its intermediate value, $y_1 = \tilde{y}_1$. The three conditions: (i) *Confusion Condition*: y_{01} is different from \tilde{y}_1 , and none of them is \perp . (ii) *Length Condition*: The size of the check set ρ_{01} is at least $T \equiv \lceil \mu \cdot m \rceil$, similarly to the Length Condition of the Check Phase above. (iii) *Consistency Condition*: For a large fraction of the indices in ρ_{01} , R_1 measures the bit value opposite to y_{01} . The tolerance parameter defining this fraction is λ , assumed to fulfill $1/2 < \lambda < 1$; that is, this condition is true if the number of indices in ρ_{01} where R_1 measures the bit opposite to y_{01} is greater or equal to $\lambda T + |\rho_{01}| - T$. Note that this Consistency Condition is less stringent than the one in the Check Phase.

We also provide a more compact and more formal definition of the protocol:

1. $S \rightarrow R_0, R_1: x_S, \sigma_S = \{\alpha \in \{1, \dots, m\} : Q_S[\alpha] = x_S x_S\}, R_j$ receive $\{x_j, \sigma_j\}$ $S: y_S = x_S$
2. $R_0: \text{if } (|\sigma_0| \geq T) \wedge (\{\alpha \in \sigma_0 : Q_0[\alpha] = x_0\} = \emptyset)$
then $y_0 = x_0$ else $y_0 = \perp$ fi
 $R_1: \text{if } (|\sigma_1| \geq T) \wedge (\{\alpha \in \sigma_1 : Q_1[\alpha] = x_1\} = \emptyset)$
then $\tilde{y}_1 = x_1$ else $\tilde{y}_1 = \perp$ fi
3. $R_0 \rightarrow R_1: y_0, \rho_0 = \sigma_0; R_1: \text{receives } (y_{01}, \rho_{01})$
4. $R_1: \text{if } (\perp \neq y_{01} \neq y_1 \neq \perp) \wedge (|\rho_{01}| \geq T) \wedge$
 $\wedge (\{|\alpha \in \rho_{01} : Q_1[\alpha] = 1 - y_{01}\} \geq \lambda T + |\rho_{01}| - T)$
then $y_1 := y_{01}$ else $y_1 = \tilde{y}_1$ fi

Protocol 1: Formal description of the Weak Broadcast protocol.

In what follows, we will use the label ‘Weak Broadcast protocol’ or simply ‘Weak Broadcast’ to denote the above protocol. The description above shows how the correct components behave. Our goal is to show that this protocol is secure for at most one faulty component in a certain range of the values of the parameters μ and λ . Roughly speaking (see below for the theorem), by security [13] we mean that the *failure probability* of the protocol, that is, the probability of not achieving weak broadcast, converges to zero as the number m of four-qubit singlet states is increased to infinity. Even though the Weak Broadcast protocol itself follows straightforwardly from earlier works [8, 9], the security theorem and its proof has not been published before, to our knowledge.

We point out an important property of the above Weak Broadcast protocol, which is shared with similar earlier protocols [7, 9, 13, 14]. There is an asymmetry between the two Receivers R_0 and R_1 . In contrast to R_0 , R_1 never sends messages, thus it has no impact on the output of the other components. In particular, if R_1 is the faulty component, then the other two components will reach consensus and hence weak broadcast is guaranteed. Therefore, from now on we do not consider the case when R_1 is faulty; we restrict our attention to three other adversary configurations, which we will refer to as ‘no faulty’, ‘S faulty’, and ‘ R_0 faulty’.

3 Security of Weak Broadcast

The Weak Broadcast protocol defined in Sec. 2 is based on Refs. [7–9, 13]. Reference [13] provides a security proof of the weak broadcast protocol proposed therein. That protocol uses entangled three-qutrit states. In this work, we apply elements from there to construct a security proof for the above-defined Weak Broadcast protocol, which is based on four-qubit singlet states. Preliminaries and the theorem are presented in this section; for the proof we refer to Appendix E.

We start by specifying our framework. We assume that there is a complete network of pairwise authenticated classical channels among the three components, S , R_0 and R_1 . We also assume that this classical network is synchronous. In particular, (1) before the protocol, the components have already agreed on a common point in time when the protocol is to be started, and (2) all components operate according to a global clock and every message sent during a clock cycle (our protocol consists of 2 clock cycles) will arrive to the receiver by the end of the cycle. Recall also that we consider three *adversary configurations*: ‘no faulty’, ‘S faulty’, and ‘ R_0 faulty’.

Here, *adversary* or *faulty component* means an *active adversary*, which might act as a properly functioning component, but it might try to undermine the protocol by sending confusing information to the other components. In fact, we assume a conscious adversary following a rational strategy. We also assume that the adversary has only local information. In other words, each random bit generated from the four-qubit singlet states by a local measurement, see Fig. 1b, is known only for the component that has performed the measurement.

We also assume that there is a quantum source that distributes the four-qubit singlet states to the components as shown in Fig. 1. This distribution happens before the measurements required for the communication steps of the Weak Broadcast protocol are performed.

After these preliminaries, we state the theorem claiming the security of Weak Broadcast:

Theorem: For any parameter pairs (μ, λ) fulfilling

$$2/9 < \mu < 1/3, \text{ and} \quad (2a)$$

$$\frac{2 + 9\mu}{18\mu} < \lambda < 1, \quad (2b)$$

the failure probability $p_f(m)$ of the Weak Broadcast protocol, defined as Protocol 1 above, converges to zero as $m \rightarrow \infty$. In particular, in all three adversary configurations, the failure probability is upper-bounded as

$$p_f(m) \leq ae^{-bm}, \quad (3)$$

with $a, b > 0$.

The proof is found in Appendix E. The parameter range where our proof guarantees security is shown in Fig. 2.

In the spirit of Refs. [7, 13], we make a remark on the complexity of the protocol: Let \mathcal{R}_q and \mathcal{B}_q be the round and bit complexities of distributing and measuring a four-qubit singlet state. Then, the Weak Broadcast protocol with a failure probability of at most $0 < \epsilon \ll 1$ requires a round complexity of $\mathcal{R} = \mathcal{R}_q + 2$, and a bit complexity of $\mathcal{B} = \mathcal{O}(\log(1/\epsilon))\mathcal{B}_q$.

A natural question is if using secure classical channels instead of authenticated ones brings any advantage. Here we reason that the answer is no. If the

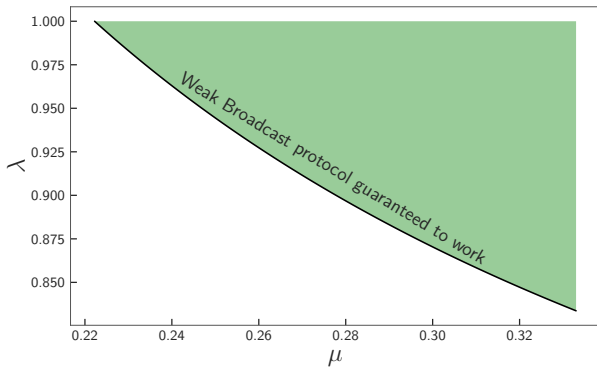


Figure 2: Parameter region where the Weak Broadcast protocol (Protocol 1) is guaranteed to work. I.e., for parameter pairs μ, λ in the green region (see Eq. (2b)) the failure probability of the Weak Broadcast protocol can be upper-bounded as $p_f(m) \leq ae^{-bm}$ with $a, b > 0$.

Sender is the faulty component, then even if it can eavesdrop on the communication between R_0 and R_1 , that has no effect on the outcome of the protocol. Furthermore, if R_0 is the faulty component, then S is correct, hence S sends the very same information to both receivers. Therefore, R_0 eavesdropping on the communication between S and R_1 does not gain extra information for R_0 . These imply that authenticated classical channels where eavesdropping is possible are just as useful in this case as secure channels.

4 Resource optimization of the Weak Broadcast protocol

In this section, our first goal is to calculate a tight upper bound for the failure probability $p_f(m)$ of the Weak Broadcast protocol. This tight upper bound will be used to estimate the resource requirements of the protocol. In particular, we will estimate the number m of singlet states that are required to achieve weak broadcast with a small failure probability, say, 5%. This analysis is similar in spirit to Ref. [11], which was carried out for a different quantum-aided multi-party communication protocol [12].

First, we derive the tight upper bound of the failure probability of the Weak Broadcast protocol as function of μ and λ . We obtain analytical results, and cross-check those against Monte-Carlo simulations. Then, we optimize the protocol in the μ - λ parameter plane. That is, we identify the parameter values that require the least resources, i.e., the smallest number of four-qubit singlet states such that the failure probability is guaranteed to be below the desired failure probability threshold.

4.1 Tight upper bound for the failure probability – analytical results

We analyse the resource requirement of the Weak Broadcast protocol as follows. We can fix the parameter values μ and λ , and the *target failure threshold* $p_{f,t}$; e.g., $\mu = 0.272$, $\lambda = 0.94$, and $p_{f,t} = 0.05$. The target failure threshold $p_{f,t} = 0.05$ means that the user can tolerate at most 5% failure probability. Then, the resource requirement of the protocol is characterized by the minimal number $m_{\min}(\mu, \lambda, p_{f,t}) \in \mathbb{Z}^+$ of four-qubit singlet states required to suppress the failure probability p_f below the threshold $p_{f,t}$.

A natural way to compute m_{\min} would be to identify the optimal adversary strategies for the S faulty and R_0 faulty configurations, i.e. those strategies that maximize the failure probability. We do not complete this task here. Instead, we identify optimal *incomplete* strategies, which are incomplete in the sense that they provide a description of the adversary’s behavior for certain Events but not for others. Using these optimal incomplete strategies, we can compute tight upper bounds on the failure probabilities $p_f(m)$, and can use those upper bounds to compute an upper bound $m_{\min, \uparrow}$ of the minimal resource requirement.

For the no faulty configuration, we derive the exact result $p_f^{(\text{nf})}$ for the failure probability shown as Eq. (24). The optimal incomplete adversary strategies for the S faulty and R_0 faulty configurations are described in App. B and App. D. The corresponding failure-probability upper bounds $p_{f, \uparrow}^{(S)}$ and $p_{f, \uparrow}^{(R)}$ are derived in App. C, and are shown in Eq. (26) and Eq. (34), respectively.

To evaluate these analytical formulas, we carry out numerical summation. The results, for a fixed value of protocol parameters, are shown in the three panels of Fig. 3, corresponding to the three adversary configurations. See caption for parameter values. All three panels show that the failure probability p_f exhibits a decreasing trend as we increase the number of four-qubit singlet states, as expected. From the data, we can read off the minimum number of states required to suppress the upper bound of the failure probability below $p_{f,t} = 0.05$: these are 143, 246 and 280, respectively, for the three adversary configurations. Our upper bound $m_{\min, \uparrow}$ of the minimal resource requirement of this protocol for $p_{f,t} = 0.05$ is the maximum of those three integers, that is, $m_{\min, \uparrow} = 280$.

4.2 Tight upper bound for the failure probability - Monte-Carlo simulations

To test our analytical calculations described in the previous subsection, here we use Monte-Carlo simulations to compute the failure probability $p_f^{(\text{nf})}$ and the failure-probability upper bounds $p_{f, \uparrow}^{(S)}$ and $p_{f, \uparrow}^{(R)}$.

In these simulations, we randomly generate, on a classical computer, the lists of correlated 4-bit strings

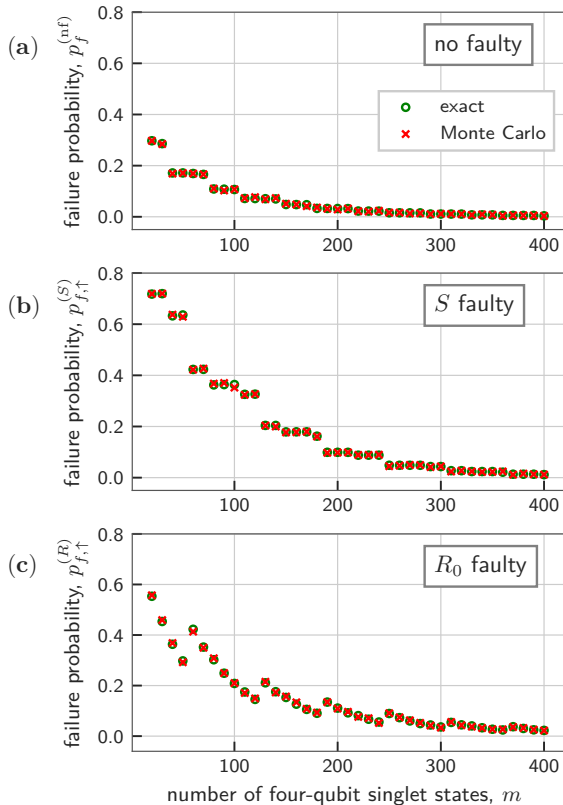


Figure 3: Failure probability of the Weak Broadcast protocol for the three adversary configurations. (a) ‘No faulty’ configuration. Exact results (green circles) and Monte-Carlo results (red crosses) for the failure probability. (b) ‘ S faulty’ configuration. (c) ‘ R_0 faulty’ configuration. In (b) and (c), the failure-probability upper bound is shown. Parameters (all panels): $\mu = 0.272$ and $\lambda = 0.94$. Simulation results are obtained from $N = 10,000$ random Events for each m . (Fig. 1b shows a single random Event for $m = 12$.)

measured by the components (i.e., the Events, see Fig. 1b). For the no faulty configuration, we evaluate for each random Event if the protocol leads to failure or weak broadcast. Then, we express the failure probability as the ratio N_f/N of the number N_f of failures and the number N of random Events. The resulting failure probability as function of m is shown as the red crosses in Fig. 3a. Data corresponds to $N = 10,000$ random Events.

For the S faulty and R_0 faulty adversary configurations, our simulation corresponds to the upper-bound calculation described in the previous subsection. For example, for the S faulty configuration, we use the optimal incomplete strategy described in App. B, and do the following steps. (1) We generate a random Event. (2) We check if the random Event is in the domain of the optimal incomplete strategy. (3a) If the Event is not in the domain, then we assume failure. (3b) If the Event is in the domain, then we evaluate the protocol such that S is acting according to its adversary strategy; this leads to either failure or weak broadcast.

(4) After repeating these steps for N different random Events, we express the failure-probability upper bound as the ratio N_f/N of the number N_f of failures and the number N of random Events. These results are shown as the red crosses in Fig. 3b and c, for the S faulty and R_0 faulty configurations, respectively. Data corresponds to $N = 10,000$ random Events.

In Fig. 3a, b, and c, the results of the Monte-Carlo simulations (red crosses) are in excellent agreement with the exact results (green circles). We interpret the small differences between the Monte-Carlo result and the exact result as statistical fluctuations of the Monte-Carlo result. For example, estimating the failure probability $p = 0.2$ from $N = 10,000$ samples has a statistical error of $\sqrt{p(1-p)/N} \approx 0.4\%$.

4.3 Optimization in the parameter space

In the previous subsections, we computed the resource requirements of the Weak Broadcast protocol for fixed values of the parameters μ and λ , and for a fixed value of the target failure threshold $p_{f,t}$. A natural next step is the optimisation of the protocol: finding the optimal parameter values λ and μ , which minimize the resource requirements, such that it is guaranteed that the failure probability is kept below the target failure threshold $p_{f,t}$. Even though we do not know the failure probabilities of the best adversary strategies, we do know tight upper bounds for the failure probability, using which we can carry out the optimization. In particular, we will minimize the function $m_{\min,\uparrow}(\mu, \lambda, p_{f,t})$ in the (μ, λ) parameter plane, with the specific value of the target failure threshold $p_{f,t} = 0.05$. We expect that future deployment of such communication protocols will require similar optimization procedures.

We perform this optimization in two steps, as follows. We use the analytical formulas discussed in section 4.1. For a rough optimization in the (μ, λ) parameter space, we consider the rectangle in Fig. 4a, and the grid of parameter pairs shown there. For each point (box) of the grid, we evaluate the failure-probability upper bound $p_{\uparrow}(m)$ for $m = 290, 300$, and color the box as blue or green, respectively, according to the minimal m within this set where $p_{f,\uparrow} < p_{f,t} = 0.05$. This procedure results in the pattern in Fig. 4a, which suggest a finer optimization focusing on the area of the dark blue rectangle of Fig. 4a. Note that grey boxes correspond to parameter values μ, λ lying outside the green zone of Fig. 2.

The results of this finer optimization are shown in Fig. 4b. The procedure is analogous to the rough optimization, with the difference that here the set of m values is $m = 270, 271, \dots, 300$. Figure 4b reveals a ‘sweet rectangle’, a rectangle of points where using 280 four-qubit singlet states is sufficient to achieve weak broadcast with at most 5% error. We emphasize that for an error threshold different from 5%,

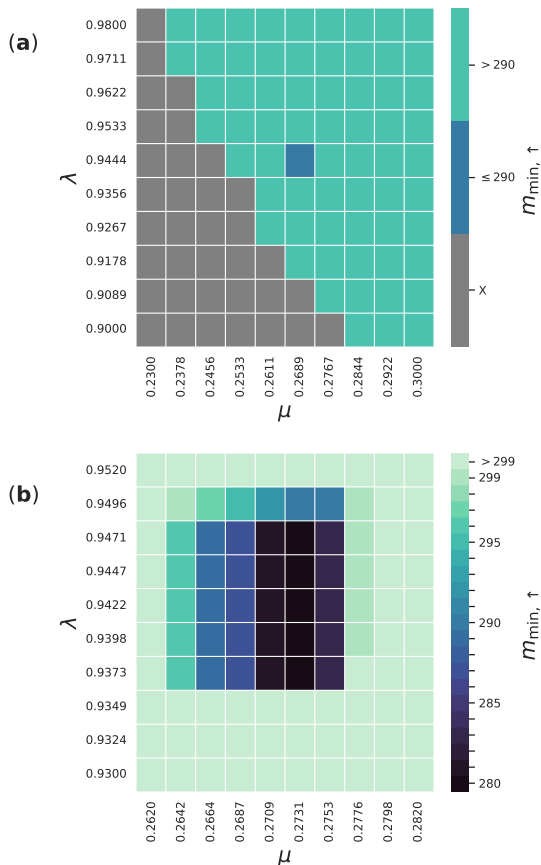


Figure 4: Optimization of the Weak Broadcast protocol in the (μ, λ) parameter space, for a fixed target failure threshold. Both panels indicate the resource requirement, i.e., number of four-qubit singlet states, to achieve failure probability below the target failure threshold $p_{\text{th}} = 0.05$. Grey boxes indicate parameter points that are outside of the guaranteed range of the protocol (i.e., are outside of the green range in Fig. 2). (a) Rough optimization. Dark pixels indicate the more economical region in the parameter space. (b) Fine optimization: zoom-in of (a). Dark pixels (‘sweet rectangle’) indicate the most economical region in the parameter space.

this optimization must be repeated, and might yield different optimal values for the parameters μ and λ .

To summarize, in this section we have outlined and implemented resource optimization procedures for the Weak Broadcast protocol. This illustrates the engineering aspects of the deployment of future quantum-aided distributed systems.

5 Four-qubit singlet state on IBM Q and IonQ devices

An important goal in the field of quantum communication is to advance the hardware technology and thereby to enable the practical deployment of quantum-aided multi-party communication [9, 15, 16]. With this goal in mind, we studied the preparation of the resource states of the Weak Broadcast pro-

tol, i.e., four-qubit singlet states, on NISQ hardware [17], namely using superconducting qubits of IBM Q [18] and trapped ion qubits of IonQ [19].

Both superconducting and trapped ion qubits have already been shown to be promising candidates for implementing quantum networks [10]. Quantum network functionalities using superconducting qubits housed in spatially separated cryogenic systems have been experimentally demonstrated recently [20], while scalable remote entanglement between trapped ions has been achieved via photon and phonon interactions [21], raising the hope that entangled states prepared on-chip on quantum processors can be used as resources in quantum networks.

In the rest of this section we identify two quantum circuits that prepare the four-qubit singlet state. To characterize the quality of our experimental state preparation, we use two quantities. The first one is the *classical fidelity* F_c , that is the fidelity between (i) the probability distribution P_{exp} of bitstrings obtained experimentally by doing computational-basis measurements, and (ii) the ideal probability distribution P_{id} of measured bitstrings, following from Eq. (1) and shown in Fig. 1c:

$$F_c = \left(\sum_{s=(0000)}^{\binom{1111}{s}} \sqrt{P_{\text{exp}}(s)P_{\text{id}}(s)} \right)^2. \quad (4)$$

The second figure of merit for state preparation is the *quantum state fidelity* F_q . We obtain this by performing quantum state tomography at the end of the circuit, reconstructing the density matrix ρ_{exp} from the experimental data, and evaluating the fidelity between this state and the ideal four-qubit singlet state $\rho_{\text{id}} = |\psi\rangle\langle\psi|$:

$$F_q = \left(\text{Tr} \sqrt{\rho_{\text{id}}^{1/2} \rho_{\text{exp}} \rho_{\text{id}}^{1/2}} \right)^2. \quad (5)$$

5.1 Circuit A

First, we need to identify a quantum circuit that prepares the four-qubit singlet state. We rely on CNOT as the two-qubit gate. On today’s noisy devices, CNOT is ‘expensive’ in the sense that its gate error is greater than the single-qubit gate errors. Therefore, a second target for our circuit search is to minimize the number of CNOTs. A third aspect is related to the incomplete connectivity of IBM Q devices: in this subsection we will pay special attention to devices that host a chain of four qubits with linear connectivity.

On linear qubit chains, we performed our measurements using Circuit A, shown in Fig. 5, which is based on the results of Ref. [22]. The construction of Ref. [22] provides a circuit with 18 CNOTs. This we have further simplified, partly by hand, partly by Qiskit’s transpiler, to a circuit containing 14 CNOTs. In the circuit we use general single-particle U gates,

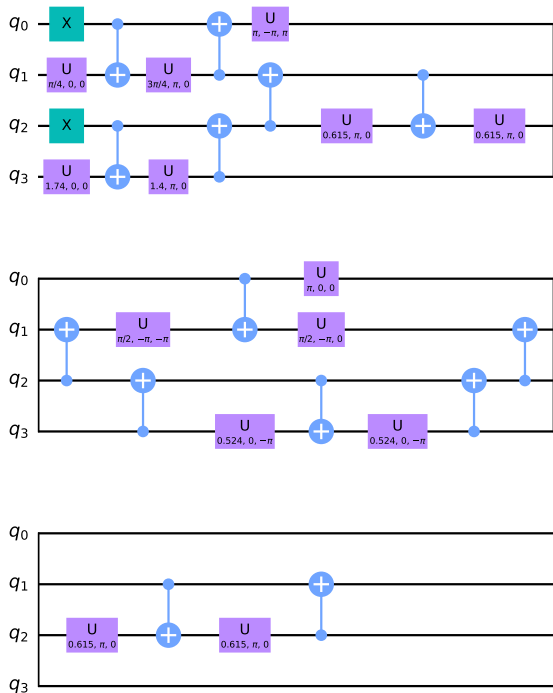


Figure 5: Circuit A, a 14-CNOT circuit that prepares the four-qubit singlet state of Eq. (1). The decimal fractions present in the circuit map to 1.7407189, 1.4008736, 0.61548365 and 0.52360105.

with the parametrization

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos(\theta/2) & -e^{i\lambda} \sin(\theta/2) \\ e^{i\phi} \sin(\theta/2) & e^{i(\lambda+\phi)} \cos(\theta/2) \end{pmatrix}. \quad (6)$$

We have run Circuit A in all four-qubit linear chains in all IBM Q devices with Quantum Volume 32 (available in 2021 April). This yields 32 qubit configurations, and the corresponding results are listed in Table 3. In Table 3, we characterize the quality of state preparation using classical and quantum fidelities introduced in Eqs. 4 and 5.

Without readout error mitigation, our best experiment was the one using the qubit chain 0-1-2-3 on `ibmq_santiago`, yielding $F_c = 0.8510$. When we correct for the readout errors using the built-in readout error mitigation procedure of Qiskit (see Refs. [23, 24] for theoretical details), then the best result is found on the same qubit chain, with $F_c = 0.9021$. For reference, the classical fidelity between a uniform bitstring distribution and the ideal bitstring distribution is $1/\sqrt{3} \approx 0.58$. Hence we conclude that the bitstring distribution provided by IBM Q hardware is much closer to the ideal one than to random noise.

The data in Table 3 show that the best-performing arrangement is the 0-1-2-3 qubit chain on `ibmq_santiago`. Without readout error mitigation, the fidelity is $F_q = 0.7435$, whereas readout error mitigation improves this to $F_q^{\text{mitig}} = 0.8116$. As a

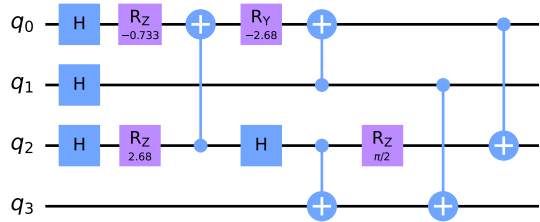


Figure 6: Circuit B, a 5-CNOT circuit that prepares the four-qubit singlet state of Eq. (1).

reference, the quantum state fidelity of the ideal four-qubit singlet state and the fully mixed four-qubit state is ≈ 0.0625 .

5.2 Circuit B

Another four-qubit circuit that prepares the four-qubit singlet state $|\psi\rangle$, is Circuit B depicted in Fig. 6, which includes only 5 CNOT gates. We found Circuit B using a combination of random circuit search and gradient descent: We assembled random circuits using CNOTs and single-qubit gates from the set $\{I, X, T, S, H\}$; searching through approximately 10^7 such instances, we selected the one whose output state had the maximal overlap with the four-qubit singlet state. Then, for the selected instance, we replaced all the one-qubit gates with generic three-parameter single-qubit gates, and performed gradient descent in this multi-dimensional parameter space, with the goal of maximizing the fidelity between the output state of the circuit and the four-qubit singlet state.

Although the decrease in the number of CNOTs with respect to Circuit A is spectacular (5 instead of 14), Circuit B did not produce better results than Circuit A in our state-preparation experiments with the IBM Q quantum processors. The reason is that Circuit B does not suit the linear qubit chain arrangements of the QV32 backends of IBM Q, hence it can only be run without qubit rearrangement on lower-quality hardware (e.g., `ibmq_16_melbourne`), which has the loop-type connectivity of Circuit B.

However, we also implemented Circuit B on IonQ's trapped ion quantum computer, which has full connectivity. Beside the better connectivity, these devices have also smaller readout and gate error rates. The experiments were run through the AWS Braket cloud service [25]. As we expected, the results were better; we achieved classical fidelities $\geq 95\%$ in all bases and a quantum fidelity (without readout error mitigation) of 88.5%, both being considerably higher than the fidelities obtained using the IBM Q quantum processors.

To conclude, we have identified two circuits, Circuit A and Circuit B, that can be used to prepare the four-qubit singlet state on IBM Q and IonQ hard-

ware. According to our best data, the circuit on IBM Q produces the desired bitstring probability distribution with classical fidelity $F_c^{\text{mitig}} = 0.9002$, and it produces the desired state with quantum state fidelity $F_q^{\text{mitig}} = 0.8116$. Correspondingly, our single run of the experiment on IonQ achieved classical fidelities $\geq 95\%$ in all bases and a quantum fidelity without readout error mitigation of 88.5%.

6 Conclusion and outlook

In this work, we introduced a two-parameter family of quantum-aided weak broadcast protocols, and proved their security in a certain range of its parameters. We computed a tight upper bound $m_{\min, \uparrow}$ on the number of resource quantum states required to suppress the failure probability of the protocol below a given target failure probability threshold. We optimized the protocol, that is, we located the parameter range where the resource requirements, as indicated by $m_{\min, \uparrow}$, are minimal. We have experimentally implemented the preparation and characterization of our four-qubit resource state on real qubits: in the superconducting quantum processors of IBM and the ion-trap quantum processors of IonQ. Our work illustrates the engineering aspects of future deployments of such protocols in practice. The methods and results of this work can be utilized in future research toward multi-party quantum communication in distributed systems.

Let us note that throughout the theoretical part of this work, we assumed that the four-qubit singlet state $|\psi\rangle$ of Eq. (1) is prepared and distributed through public quantum channels to the three components of the distributed system. The role of these entangled states is to provide random bits that are correlated among the different components. The same functionality cannot be substituted by a source of correlated classical random bits, distributed through public channels. In this case, the adversary can eavesdrop on the classical channel transmitting the random bits from the source to the components, and hence the criterion that each component knows only its own measurement outcomes can be violated. In the quantum case, considered in this work, the eavesdropping can be detected by sacrificing a fraction of the four-qubit singlet state in a distribute-and-test procedure of the protocol, similar to the procedures described in preceding work [8, 9, 14]. We leave it for future work to elaborate the quantitative details of such a distribute-and-test procedure.

The above observation points toward further important follow-up tasks to this work. For example, an attack against the protocol could be based on the adversary gaining control over the source, or over the measurement devices generating the correlated random bits. A further difficulty is posed by physical errors of the source, the quantum channels, and the measurement devices, as highlighted by our experiments

presented in Sec. 5. We anticipate that the effect of these imperfections on the protocol will be studied in a framework similar to that of device-independent quantum key distribution [26, 27]. Some of these issues have already been discussed [7–9], but a quantitative study is, to our knowledge, still missing.

Acknowledgments

We acknowledge useful discussions with T. Coopmans, M. Farkas, M. Fitzi, B. T. Gard, T. Kriváchy, and A. Pereszlényi. This research was supported by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office (NKFIH) within the Quantum Information National Laboratory of Hungary, the Quantum Technology National Excellence Program (Project No. 2017-1.2.1-NKP-2017-00001) and the Thematic Excellence Programmes TKP2020-NKA-06 and TKP2020 IES (Grant No. BME-IE-NAT), under the auspices of the Ministry for Innovation and Technology, and by NKFIH through the OTKA Grants FK 124723, FK 132146, K 124351. Furthermore, we acknowledge the use of the IBM Q for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Q team. We acknowledge the access to advanced services provided by the IBM Quantum Researchers Program.

References

- ¹M. Pease, R. Shostak, and L. Lamport, “Reaching agreement in the presence of faults”, *J. ACM* **27**, 228–234 (1980) DOI: [10.1145/322186.322188](https://doi.org/10.1145/322186.322188).
- ²L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem”, *ACM Trans. Program. Lang. Syst.* **4**, 382–401 (1982) DOI: [10.1145/357172.357176](https://doi.org/10.1145/357172.357176).
- ³J. Garay and Y. Moses, “Fully Polynomial Byzantine Agreement for Processors in Rounds”, *SIAM Journal on Computing* **27**, DOI: [10.1137/S0097539794265232](https://doi.org/10.1137/S0097539794265232) (1999) DOI: [10.1137/S0097539794265232](https://doi.org/10.1137/S0097539794265232).
- ⁴M. Castro, B. Liskov, et al., “Practical byzantine fault tolerance”, in Osdi, Vol. 99, 1999 (1999), pp. 173–186.
- ⁵D. Ongaro and J. Ousterhout, “In Search of an Understandable Consensus Algorithm”, in Proceedings of the 2014 usenix conference on usenix annual technical conference, USENIX ATC’14 (2014), 305–320.
- ⁶L. Lamport, “The Part-Time Parliament”, *ACM Trans. Comput. Syst.* **16**, 133–169 (1998) DOI: [10.1145/279227.279229](https://doi.org/10.1145/279227.279229).

- ⁷M. Fitz, N. Gisin, U. M. Maurer, and O. v. Rotz, “Unconditional Byzantine Agreement and Multi-Party Computation Secure against Dishonest Minorities from Scratch”, in Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology, EUROCRYPT ’02 (2002), 482–501.
- ⁸A. Cabello, “Solving the liar detection problem using the four-qubit singlet state”, *Phys. Rev. A* **68**, 012304 (2003) DOI: [10.1103/PhysRevA.68.012304](https://doi.org/10.1103/PhysRevA.68.012304).
- ⁹S. Gaertner, M. Bourennane, C. Kurtsiefer, A. Cabello, and H. Weinfurter, “Experimental Demonstration of a Quantum Protocol for Byzantine Agreement and Liar Detection”, *Phys. Rev. Lett.* **100**, 070504 (2008) DOI: [10.1103/PhysRevLett.100.070504](https://doi.org/10.1103/PhysRevLett.100.070504).
- ¹⁰P. Pathumsoot, T. Matsuo, T. Satoh, M. Hajdušek, S. Suwanna, and R. Van Meter, “Modeling of measurement-based quantum network coding on a superconducting quantum processor”, *Phys. Rev. A* **101**, 052301 (2020) DOI: [10.1103/PhysRevA.101.052301](https://doi.org/10.1103/PhysRevA.101.052301).
- ¹¹M. A. Taherkhani, K. Navi, and R. V. Meter, “Resource-aware system architecture model for implementation of quantum aided byzantine agreement on quantum repeater networks”, *Quantum Science and Technology* **3**, 014011 (2017) DOI: [10.1088/2058-9565/aa9bb1](https://doi.org/10.1088/2058-9565/aa9bb1).
- ¹²M. Ben-Or and A. Hassidim, “Fast Quantum Byzantine Agreement”, in *Proceedings of the thirty-seventh annual acm symposium on theory of computing*, STOC ’05 (2005), 481–485, DOI: [10.1145/1060590.1060662](https://doi.org/10.1145/1060590.1060662).
- ¹³M. Fitz, “Generalized communication and security models in Byzantine agreement”, Reprint as vol. 4 of ETH Series in Information Security and Cryptography, ISBN 3-89649-853-3, Hartung-Gorre Verlag, Konstanz, 2003, PhD thesis (ETH Zurich, Mar. 2003).
- ¹⁴M. Fitz, N. Gisin, and U. Maurer, “Quantum solution to the byzantine agreement problem”, *Phys. Rev. Lett.* **87**, 217901 (2001) DOI: [10.1103/PhysRevLett.87.217901](https://doi.org/10.1103/PhysRevLett.87.217901).
- ¹⁵S. Wehner, D. Elkouss, and R. Hanson, “Quantum internet: a vision for the road ahead”, *Science* **362**, eaam9288 (2018) DOI: [10.1126/science.aam9288](https://doi.org/10.1126/science.aam9288).
- ¹⁶M. Pompili, S. L. N. Hermans, S. Baier, H. K. C. Beukers, P. C. Humphreys, R. N. Schouten, R. F. L. Vermeulen, M. J. Tiggeleman, L. dos Santos Martins, B. Dirkse, S. Wehner, and R. Hanson, “Realization of a multinode quantum network of remote solid-state qubits”, *Science* **372**, 259–264 (2021) DOI: [10.1126/science.abg1919](https://doi.org/10.1126/science.abg1919).
- ¹⁷J. Preskill, “Quantum computing in the nisq era and beyond”, *Quantum* **2**, 79 (2018).
- ¹⁸*IBM Quantum*, <https://quantum-computing.ibm.com/>, 2021.
- ¹⁹C. Monroe, “Ionq quantum computers: clear to scale”, in Aps march meeting abstracts, Vol. 2021 (2021), P10–002.
- ²⁰P. Magnard, S. Storz, P. Kurpiers, J. Schär, F. Marxer, J. Lütolf, T. Walter, J.-C. Besse, M. Gabureac, K. Reuer, A. Akin, B. Royer, A. Blais, and A. Wallraff, “Microwave quantum link between superconducting circuits housed in spatially separated cryogenic systems”, *Phys. Rev. Lett.* **125**, 260502 (2020) DOI: [10.1103/PhysRevLett.125.260502](https://doi.org/10.1103/PhysRevLett.125.260502).
- ²¹D. Hucul, I. V. Inlek, G. Vittorini, C. Crocker, S. Debnath, S. M. Clark, and C. Monroe, “Modular entanglement of atomic qubits using photons and phonons”, *Nature Physics* **11**, 37–42 (2015).
- ²²B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, “Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm”, *npj Quantum Information* **6**, DOI: [10.1038/s41534-019-0240-1](https://doi.org/10.1038/s41534-019-0240-1) (2020) DOI: [10.1038/s41534-019-0240-1](https://doi.org/10.1038/s41534-019-0240-1).
- ²³Y. Chen, M. Farahzad, S. Yoo, and T.-C. Wei, “Detector tomography on ibm quantum computers and mitigation of an imperfect measurement”, *Physical Review A* **100**, 052315 (2019).
- ²⁴F. B. Maciejewski, Z. Zimborás, and M. Oszmaniec, “Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography”, *Quantum* **4**, 257 (2020).
- ²⁵C. Gonzalez, “Cloud based qc with amazon braket”, *Digitale Welt* **5**, 14–17 (2021).
- ²⁶A. Acín, N. Brunner, N. Gisin, S. Massar, S. Pironio, and V. Scarani, “Device-independent security of quantum cryptography against collective attacks”, *Phys. Rev. Lett.* **98**, 230501 (2007) DOI: [10.1103/PhysRevLett.98.230501](https://doi.org/10.1103/PhysRevLett.98.230501).
- ²⁷F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, “Secure quantum key distribution with realistic devices”, *Reviews of Modern Physics* **92**, DOI: [10.1103/revmodphys.92.025002](https://doi.org/10.1103/revmodphys.92.025002) (2020) DOI: [10.1103/revmodphys.92.025002](https://doi.org/10.1103/revmodphys.92.025002).

S	R_0	R_1	no faulty	S faulty	R_0 faulty
0	0	0	✓	✓	✓
0	0	1	×	×	×
0	1	0	×	×	✓
0	1	1	×	✓	×
1	0	0	×	✓	×
1	0	1	×	×	✓
1	1	0	×	×	×
1	1	1	✓	✓	✓

Table 1: Truth table of broadcast with $n = 3$ components, resilient up to $t = 1$ faulty component. Column 1 is the input and output bit of the sender (S), column 2 (3) is the output bit of receiver R_0 (R_1). Columns 4, 5, 6 indicate whether the list of output bits and the faulty configuration together satisfies the conditions of broadcast or not. The faulty configuration ' R_1 faulty' is not shown as it is analogous to R_0 faulty.

A Broadcast, weak broadcast, and their truth tables

In Sec. 2, we introduced the broadcast and the weak broadcast functionalities of multi-party communication protocols. Here, we recall their definitions [13]. Furthermore, we show their tabular representations ('truth tables'), for the case of $n = 3$ components and at most $t = 1$ faulty component.

Definition 1. (broadcast) A protocol among n components such that one distinct component S (the Sender) holds an input value $x_S \in \{0, 1\}$, and all other components (the receivers) eventually decide on an output value in $\{0, 1\}$ is said to achieve **broadcast** if the protocol guarantees the following conditions:

Validity: if the Sender is correct then all correct components decide on $y = x_S$.

Consistency: all correct components decide on the same output.

The truth table of the broadcast functionality is shown in Table 1, for the special case when the number of components is $n = 3$, and resilience is expected up to $t = 1$ faulty component.

In the definition of the second functionality, weak broadcast, the conditions of broadcast are relaxed to some extent, and in addition to 0 and 1, a third output value called 'abort' (\perp) is also allowed. As discussed in the main text, the weak broadcast functionality is important because it can be used as a building block to achieve broadcast [13].

Definition 2. (weak broadcast) A protocol among n components such that one distinct component S (the Sender) holds an input value $x_S \in \{0, 1\}$ and all other components eventually decide on an output value in $\{0, 1, \perp\}$ is said to achieve **weak broadcast** if the protocol guarantees the following conditions:

Validity: if the Sender is correct then all correct components decide on $y = x_S$.

Consistency: if any correct component decides on an output $y \in \{0, 1\}$ then all correct components decide on a value in $\{y, \perp\}$; that is, either they choose the value y or choose abort.

The truth table of the weak broadcast functionality is shown in Table 1, for the special case when the number of components is $n = 3$, and resilience is expected up to $t = 1$ faulty component. In the main text, this special case is denoted by WBC(3,1).

B Adversary strategies against the Weak Broadcast protocol

In this section, we define adversary strategies, and propose an S faulty strategy and an R_0 faulty strategy. We prove that the strategies proposed here are optimal, i.e., they maximise the failure probability, in App. D.

As a preliminary for this section, recall that in Sec. 2, we defined the Elementary Events as the possible correlated random outcomes of the measurement of the four-qubit singlet state $|\psi\rangle$ of Eq. (1). There are six such outcomes, 0011, 1100, 1010, 0101, 1001, and 0110. It follows from Eq. (1) that the probabilities of 0011 and 1100 are $1/3$, whereas the probabilities of the remaining four bitstrings are $1/12$.

Furthermore, we have denoted the number of four-qubit singlet states, available for the weak broadcast of a single bit from S , as m . In Sec. 2, we defined an Event as the random configuration of the m bitstrings measured

S	R_0	R_1	no faulty	S faulty	R_0 faulty
0	0	0	✓	✓	✓
0	0	1	×	×	×
0	0	⊥	×	✓	×
0	1	0	×	×	✓
0	1	1	×	✓	×
0	1	⊥	×	✓	×
0	⊥	0	×	✓	✓
0	⊥	1	×	✓	×
0	⊥	⊥	×	✓	×
1	0	0	×	✓	×
1	0	1	×	×	✓
1	0	⊥	×	✓	×
1	1	0	×	×	×
1	1	1	✓	✓	✓
1	1	⊥	×	✓	×
1	⊥	0	×	✓	×
1	⊥	1	×	✓	✓
1	⊥	⊥	×	✓	×

Table 2: Truth table of WBC(3,1), that is, weak broadcast with $n = 3$ components, resilient up to $t = 1$ faulty component. Column 1 is the input and output bit of the sender (S), column 2 (3) is the output value of receiver R_0 (R_1). Columns 4, 5, 6 indicate whether the list of output values and the faulty configuration together satisfies the conditions of weak broadcast or not. The faulty configuration ' R_1 faulty' is not shown as it is analogous to R_0 faulty.

on the m distributed four-qubit singlet states by the three components. Fig. 7a shows an example of an Event for $m = 12$ (actually, this is the same Event as in Fig. 1b). The first column is the index of the singlet state, and columns 2-4 show the qubit measurement outcomes (Elementary Events) obtained by the three components. Note that the number of possible different Events is 6^m , and for a given m , it is straightforward to calculate the occurrence probability of each possible Event. For example, the Event in Fig. 7 occurs with probability $(1/3)^8(1/12)^4$.

B.1 S faulty

For concreteness, and without the loss of generality, here we assume that the faulty S is a conscious adversary, which wants to reach failure of Weak Broadcast such that receiver R_0 outputs $y_0 = 0$ and receiver R_1 outputs $y_1 = 1$. Hence, the message sent by S to R_0 is $x_0 = 0$, whereas the message sent by S to R_1 is $x_1 = 1$.

Let us first formalize what we mean by an adversary strategy (of S). To this end, we introduce two natural classifications of the Events; a finer, *global* classification, and a coarser, *local* one. Here, the term 'local' refers to the property that the classification is based only on the information available to the adversary sender S .

We define the *global count list* of an Event as the list of the six non-negative integers that count the frequencies of the six different measurement outcomes of the Event. The set of the global count lists will be denoted as GCL. That is, an element of GCL is

$$g = (g_1, g_2, g_3, g_4, g_5, g_6) = (m_{0011}, m_{0101}, m_{0110}, m_{1001}, m_{1010}, m_{1100}). \quad (7)$$

Note that for a certain Event, the adversary S does not know the corresponding global count list g , as S knows only the first two bit values of each row of the Event.

Clearly, for any $g \in \text{GCL}$ it holds that $\sum_{i=1}^6 g_i = m$. Furthermore, we will denote the counting function that associates an element of GCL to an Event as

$$G : \text{Events} \rightarrow \text{GCL}. \quad (8)$$

Note that this function is not injective, since in general, multiple Events are mapped to a single global count list $g \in \text{GCL}$. In fact, for any $g \in \text{GCL}$, the number of Events that are mapped to g via G is

$$|G^{-1}(g)| = \binom{m}{g_1, g_2, g_3, g_4, g_5, g_6}, \quad (9)$$

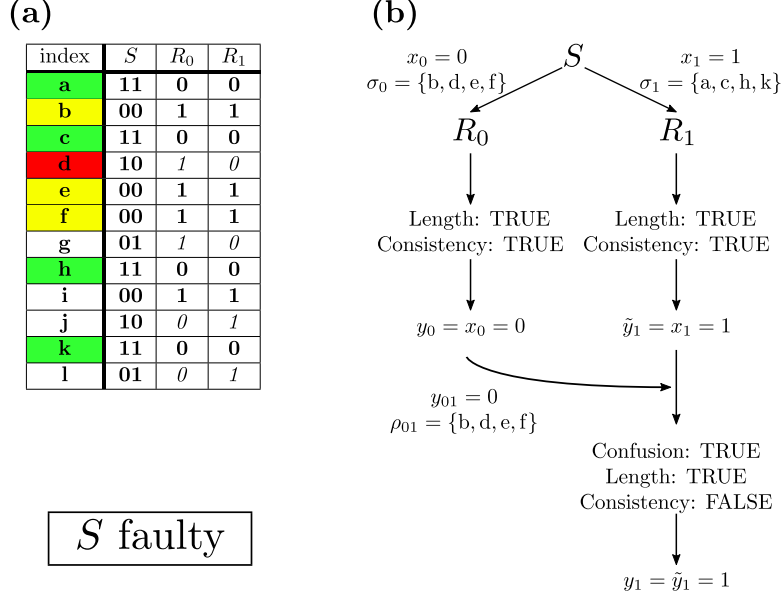


Figure 7: An example for a Sender (S) adversary strategy, which causes failure of Weak Broadcast in this particular random instance (Event). (a) Random bits resulting from the measurement of $m = 12$ four-qubit singlet states. Boldface (italic) denotes the information possessed by (hidden from) S . (b) Flowchart of the communication and decision events in case of a faulty S and correct R_0 and R_1 , for the Event in (a) and the adversary strategy $\zeta = (3, 1, 0; 0, 0, \ell_3)$. For this random instance, the strategy S leads to failure of Weak Broadcast, as the two receivers have different output values, $y_0 \neq y_1$.

where the bracket on the right hand side denotes the multinomial coefficient.

An adversary strategy of the sender S describes how S uses its own local information to act during the protocol. More precisely, the strategy is a procedure, according to which the adversary S selects the check sets σ_0 and σ_1 , to be sent to R_0 and R_1 , respectively, in the Invocation Phase. Locally, the sender S cannot distinguish between the outcomes 0101 and 0110, as S knows only the first two bits of these bitstrings. Similarly, S cannot distinguish between the outcomes 1010 and 1001. These imply that the basis of an adversary strategy of S is provided by the four non-negative integers $(g_1, g_2 + g_3, g_4 + g_5, g_6) = (m_{0011}, m_{0101} + m_{0110}, m_{1001} + m_{1010}, m_{1100})$.

For our purposes, it is sufficient to use an even coarser local classification, which does not distinguish between the local bit pairs 01 and 10 of the sender S . This coarser classification is defined by the map

$$\mathcal{L}_S : \text{GCL} \rightarrow \text{LCL}_S, g \mapsto \ell \equiv (\ell_1, \ell_2, \ell_3) \equiv (g_1, g_2 + g_3 + g_4 + g_5, g_6). \quad (10)$$

Here, LCL_S is the set of *local count lists* for the S faulty scenario, which is implicitly defined as the range of \mathcal{L}_S , containing triplets of non-negative integers, such that $\ell_1 + \ell_2 + \ell_3 = m$. The term *local count list* hence refers to the integer triplets $\ell \in \text{LCL}_S$.

Because of the sum rule $\ell_1 + \ell_2 + \ell_3 = m$, a local count list ℓ is represented by two of its elements, e.g., ℓ_1 and ℓ_3 , and hence the set LCL_S can be visualised in a two-dimensional plot. This is illustrated for the case $m = 12$ in Fig. 8, where each non-grey square (i.e., orange and blue squares) represents a local count list.

The adversary strategy of S describes how the behavior of S is derived from the local information S has. For our purposes, it is sufficient to regard the local count list ℓ associated to the Event as the local information possessed by S . We define an adversary strategy of S as a function ζ_S that maps the local count list of S to a list of 6 non-negative integers,

$$\zeta_S : \text{LCL}_S \rightarrow (\mathbb{Z}_0^+)^6, \ell \mapsto (k_{0011}^{(0)}, k_{\text{mixed}}^{(0)}, k_{1100}^{(0)}; k_{0011}^{(1)}, k_{\text{mixed}}^{(1)}, k_{1100}^{(1)}). \quad (11)$$

Here, $k_{0011}^{(0)} \leq \ell_1$ indicates how many of the 0011 indices does S include in the check set σ_0 sent to R_0 in the Invocation phase. Similarly, $k_{\text{mixed}}^{(0)} \leq \ell_2$ indicates the number of mixed-outcome (01XX and 10XX) indices to be included in σ_0 , and $k_{1100}^{(1)} \leq \ell_3$ indicates the number of 1100 indices to be included in σ_1 , etc.

Our purpose in this work is to analyse the failure probability of certain strategies. For this purpose, it is sufficient to define a strategy by the numbers above: how many indices from each class are included in each check set. It is *not* necessary to specify *which* indices are included from each class, because the failure probability is independent of the specific choice. Nevertheless, in the examples considered in this work, we assume that S includes the ‘smallest’ indices from each class, in alphabetical order of the indices (see Fig. 7).

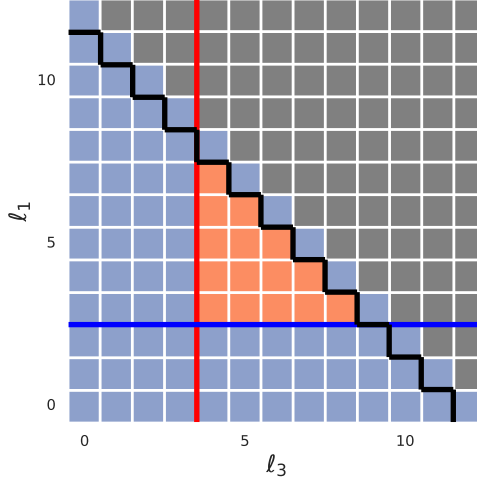


Figure 8: Local count lists and the domain of an incomplete strategy of an adversary Sender (S faulty configuration). Each non-grey square depicts a local count list in case of $m = 12$. Orange region depicts the domain $D(\zeta)$ of the optimal incomplete adversary strategy ζ , for parameter values $\mu = 0.272$, and $\lambda = 0.94$. The strategy ζ is defined in Eq. (12), whereas the domain $D(\zeta)$ is defined via Eqs. (15), (16), and (17), depicted as the blue, black, and red lines, respectively. Blue region depicts those local count lists where the incomplete strategy ζ is not defined.

We call a strategy, as defined in Eq. (11), a *complete strategy*, if its domain is the complete LCL $_S$; otherwise, we call it an *incomplete strategy*. For a complete strategy, the failure probability p_f is a well-defined real number $p_f \in [0, 1]$. From now on, we will deal with incomplete strategies, and hence we will not be able to compute the specific failure probability; however, we can derive a relatively tight upper bound for the failure probability.

In what follows, we will often consider sequences of strategies: such a strategy sequence is a function that specifies a strategy for each m . A natural notation for a strategy in such a sequence $\zeta_{S,m}$; note that we will often suppress the m , and call such a sequence of strategies simply as a strategy. In general, for different strategies in a given sequence, the domain and the failure probability (or the failure-probability upper bound, for incomplete strategies) are different.

In this work, we focus on a sequence of optimal incomplete adversary strategies of S . This is given as

$$\zeta_S(\ell_1, \ell_2, \ell_3) = (T - Q, Q, 0; 0, 0, \ell_3), \quad (12)$$

where

$$T = \lceil \mu m \rceil, \quad (13)$$

$$Q = T - \lceil \lambda T \rceil + 1, \quad (14)$$

with $\lceil \cdot \rceil$ denoting the ceiling function. E.g., for $\mu = 0.26$, $\lambda = 0.94$, and $m = 1200$, these are $T = 312$ and $Q = 19$. Recall that T , originally defined in the Weak Broadcast protocol in Sec. 2, is the minimal check set length that satisfies the Length Conditions of the Weak Broadcast protocol. Integer Q is the minimal number of inconsistent indices in ρ_{01} that implies the violation of the Consistency Condition of the Cross-check Phase, in the case when the length of the check set is $|\rho_{01}| = T$.

Note that Eq. (12) contains two implicit restrictions on the domain of ζ_S . First, the adversary S can include $T - Q$ indices of 0011 outcomes in its check set σ_0 only if the number ℓ_1 of 0011 outcomes is sufficient, that is, if

$$T - Q \leq \ell_1. \quad (15)$$

Second, the adversary S can include Q indices of mixed outcomes in its check set σ_0 only if the number ℓ_2 of mixed outcomes is sufficient, that is, if

$$Q \leq \ell_2. \quad (16)$$

Besides these two restrictions, we further restrict the domain of ζ by requiring

$$T \leq \ell_3. \quad (17)$$

The above 3 conditions define the domain $D(\zeta)$ of the strategy ζ . With this extra condition, the incomplete strategy ζ of Eq. (12) is optimal in its domain $D(\zeta)$, which will be proven in App. D.1. The domain $D(\zeta)$ is

shown as the orange region in Fig. 8, and the conditions Eqs. (15), (16), and (17) are depicted as the blue, black, and red lines in Fig. 8, respectively.

How can the strategy ζ in Eq. (12) lead to failure? This is exemplified by a specific instance of the Weak Broadcast protocol in Fig. 7, for $m = 12$. To be specific, we consider a Weak Broadcast protocol with parameter values $\mu = 0.26$ and $\lambda = 0.94$. The adversary strategy derived from the optimal strategy of Eq. (12) using these values of μ , λ and m can be written as

$$\zeta = (3, 1, 0; 0, 0, \ell_3), \quad (18)$$

since $T = 4$ and $Q = 1$.

The table in Fig. 7a shows an Event for $m = 12$. The local information of S , i.e., the information S possesses, is denoted by boldface characters in Fig. 7a: S knows the indices of the four-qubit singlet states used in the protocol, it knows its own measurement outcomes (bit-pairs), and it also knows the measurement outcomes of R_0 and R_1 for those rows where S measured 00 or 11. The information S does not possess is denoted by italic characters in Fig. 7a.

Based on the local information S has, it classifies the indices $\alpha \in \{1, \dots, m\} \equiv \{a, b, \dots, l\}$ of the Event into the three classes defined above: 0011, mixed, 1100. In the example of Fig. 7a, these classes can be identified as the corresponding check sets: 0011 = {b, e, f, i}, mixed = {d, g, j, l}, and 1100 = {a, c, h, k}. This implies $m_{0011} = m_{\text{mixed}} = m_{1100} = 4$, i.e., $\ell_1 = \ell_2 = \ell_3 = 4$. This local count list $\ell = (4, 4, 4)$ is in the domain, defined by Eqs. (15), (16), (17), of the incomplete strategy ζ in Eq. (18). Hence S can follow the incomplete strategy ζ_S , which results in the communication and decision steps depicted in Fig. 7b. There, the receivers R_0 and R_1 follow the Weak Broadcast protocol, and their output bit values are $y_0 = 0$ and $y_1 = 0$, respectively, implying failure of the Weak Broadcast protocol.

The above example sheds light on why the strategy ζ_S in Eq. (12) is efficient (in fact, optimal in its domain) in achieving failure. In more general terms, its efficiency is reasoned as follows. According to this strategy, the adversary sender S tries to send a check set σ_0 such that R_0 finds it convincingly long (that is, the Length Condition of the Check Phase is satisfied) and consistent (that is, the Consistency Condition of the Check Phase is satisfied); but when sent over to R_1 as ρ_{01} , then R_1 finds it inconsistent (Consistency Condition of the Cross-Check Phase is violated), resulting in failure via $y_0 = 0$ and $y_1 = 1$. This is achieved by the strategy ζ_S of Eq. (12), if *each* of the Q rows chosen from the mixed class by S contain bit 1 at R_0 and bit 0 at R_1 . If at least a single one of these rows contains bit 0 at R_0 and bit 1 at R_1 , then R_0 finds the check set σ_0 inconsistent with the data bit $x_0 = 0$ and hence produces an output $y_0 = \perp$, leading to Weak Broadcast (see Table 2, row 8 or row 17).

B.2 R_0 faulty

For concreteness, and without the loss of generality, here we assume that the correct sender S sends the message $x_S = x_0 = x_1 = 0$ to the receivers. Furthermore, we assume that the faulty R_0 is a conscious adversary, which wants to reach failure of Weak Broadcast such that the correct R_1 has an output bit value $y_1 = 1$ that conflicts with the output bit value of $x_S = 0$ of S .

We start by formalizing what we mean by an adversary strategy of R_0 . Similarly to the S faulty configuration, our starting point is the set GCL of global count lists, see Eq. (7). The adversary strategy of the receiver R_0 describes how R_0 uses its own local information to act during the protocol. Then, locally, the receiver R_0 can distinguish between three types of outcomes. First, if the third bit of the outcome, i.e., the bit R_0 has measured, carries the value ‘1’, and the index of the outcome is sent from S to R_0 in the check set σ_0 , then R_0 can identify the outcome as 0011. Second, if the third bit of the outcome carries the value ‘1’, and the index of the outcome is not included in σ_0 , then R_0 identifies the outcome as ‘either 0110 or 1010’. In any other case, i.e., if the third bit of the outcome carries the value of ‘0’, then R_0 identifies the outcome as ‘either 0101, or 1001, or 1100’. It is natural to denote the frequency of each type of outcome in a given Events as m_{0011} , $m_{\text{XX}10}$, and $m_{\text{XX}0\text{X}}$.

This threefold classification of the Events according to R_0 ’s local information is therefore defined by the map

$$\mathcal{L}_R : \text{GCL} \rightarrow \text{LCL}_R, g \mapsto \ell \equiv (\ell_1, \ell_2, \ell_3) \equiv (m_{0011}, m_{\text{XX}10}, m_{\text{XX}0\text{X}}) \equiv (g_1, g_3 + g_5, g_2 + g_4 + g_6). \quad (19)$$

Here, LCL_R is the set of local count lists for the R_0 faulty scenario, which is implicitly defined as the range of \mathcal{L}_R , containing triplets of non-negative integers, such that $\ell_1 + \ell_2 + \ell_3 = m$. The term *local count list* hence refers to the integer triplets $\ell \in \text{LCL}_R$.

Because of the sum rule $\ell_1 + \ell_2 + \ell_3 = m$, a local count list ℓ is represented by two of its elements, e.g., ℓ_1 and ℓ_2 , and hence the set LCL_R can be visualised in a two-dimensional plot. This is illustrated for the case $m = 12$ in Fig. 9, where each non-grey square represents a local count list.

The adversary strategy of R_0 describes the procedure to assemble the check set ρ_{01} , based on the local information R_0 has. Hence, we define an adversary strategy of R_0 as a function ζ_R that maps the local count

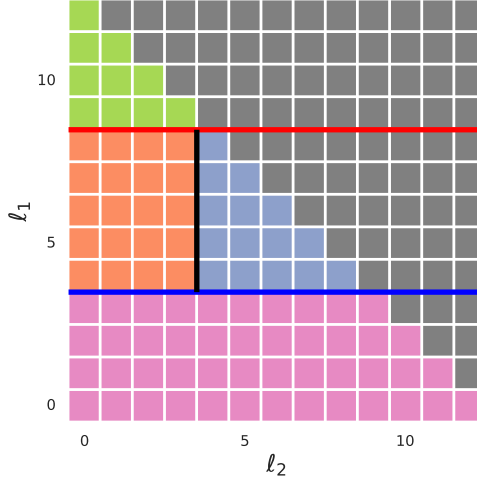


Figure 9: Local count lists and the domain of an incomplete optimal strategy of an adversary receiver R_0 (R_0 faulty configuration). Each non-grey square depicts a local count list in the case of $m = 12$. The union of orange, blue and pink regions depicts the domain $D(\zeta)$ of the optimal incomplete strategy ζ , for parameter values $\mu = 0.272$ and $\lambda = 0.94$. The strategy ζ is defined in Eq. (21), whereas the domain is defined by Eq. (23). Green region depicts local count list where the incomplete strategy ζ is not defined.

list of R_0 to a list of 3 non-negative integers:

$$\zeta_R : \text{LCL}_R \rightarrow (\mathbb{Z}_0^+)^3, \ell \mapsto (k_{0011}, k_{XX10}, k_{XX0X}). \quad (20)$$

Here, $k_{0011} \leq \ell_1$ indicates how many of the 0011 indices does R_0 include in the check set ρ_{01} sent to R_1 in the Cross-calling Phase. Similarly, $k_{XX10} \leq \ell_2$ indicates the number of XX10 indices to be included in ρ_{01} , and $k_{XX0X} \leq \ell_3$ indicates the number of XX0X indices to be included in ρ_{01} .

Similarly to the case of the S adversary strategy discussed above, here it is also sufficient to define a strategy by the ‘ k ’ numbers above in Eq. (20): how many indices from each class are included in each check set. It is *not* necessary to specify *which* indices are included from each class, because the failure probability is independent of the specific choice. Nevertheless, in the examples considered in this work, we assume that R_0 includes the ‘smallest’ indices from each class, in alphabetical order of the indices (see Fig. 10).

We call a strategy, as defined in Eq. (20), a complete strategy, if its domain is the complete LCL_R ; otherwise, we call it an incomplete strategy. From now on, we deal with incomplete strategies, and hence we will not be able to compute the specific failure probability; however, we will derive an upper bound for the failure probability.

We consider the incomplete strategy defined as

$$\zeta_R(\ell_1, \ell_2, \ell_3) = (0, \ell_2, n_{\min}), \quad (21)$$

where n_{\min} is the smallest non-negative integer such that $\ell_2 + n_{\min} \geq T$. That is,

$$n_{\min} = \begin{cases} T - \ell_2 & \text{if } \ell_2 < T \\ 0 & \text{if } \ell_2 \geq T. \end{cases} \quad (22)$$

The above definition of ζ_R contains an implicit assumption on the domain of ζ_R : the strategy can be followed only if the value of n_{\min} , as defined in Eq. (22), is not greater than ℓ_3 . More precisely, if $\ell_2 < T$, then $T - \ell_2 \leq \ell_3$ must hold. Formulating this in terms of ℓ_1 and ℓ_2 : if $\ell_2 < T$, then $\ell_1 \leq m - T$ must hold. As $\ell_2 \geq T$ implies $\ell_1 \leq m - T$, we conclude that the domain of the above strategy is defined by

$$\ell_1 \leq m - T. \quad (23)$$

In Fig. 9, this domain $D(\zeta_R)$ is illustrated for a specific choice of parameters (see caption). There, the domain $D(\zeta_R)$ of the strategy is shown as the union of the orange, blue and pink regions, whereas the green region is the complement $\bar{D}(\zeta_R)$ of the domain. Furthermore, the condition (23) is depicted as the red line.

We claim that the incomplete strategy ζ_R is optimal on its domain. This will be proven in App. D.2.

How can the strategy ζ_R of Eq. (21) lead to failure? This is exemplified by a specific instance of the Weak Broadcast protocol in Fig. 10, for $m = 12$. To be specific, we consider a Weak Broadcast protocol with parameter

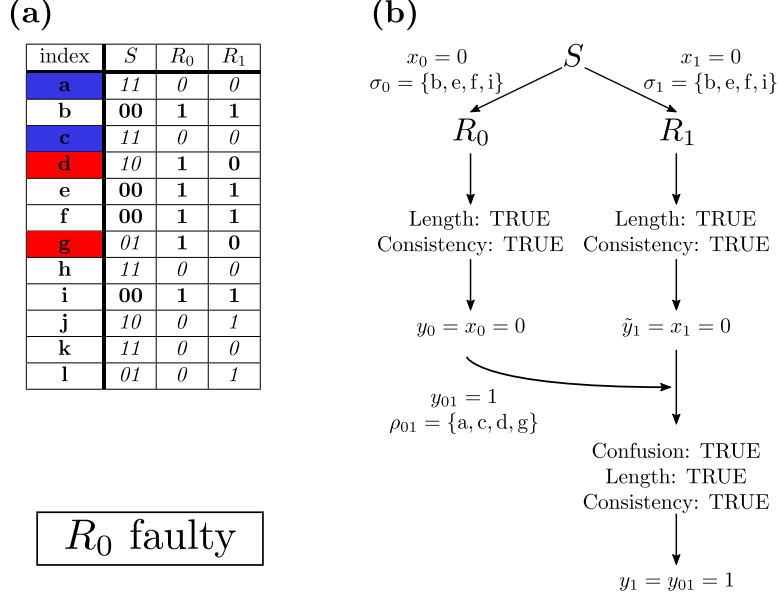


Figure 10: An example for a Receiver (R_0) adversary strategy, which causes failure of Weak Broadcast in this particular random instance (Event). (a) Random bits resulting from the measurement of $m = 12$ four-qubit singlet states. Boldface (italic) denotes the information possessed by (hidden from) R_0 . (b) Flowchart of the communication and decision steps in case of a faulty R_0 and correct S and R_1 , for the Event in (a) and the adversary strategy $S = (0, \ell_2, n_{\min})$, see Eq. (21). For this random Event, the strategy ζ_R leads to failure of Weak Broadcast, as the two receivers have different outputs values, $y_0 \neq y_1$.

values $\mu = 0.26$ and $\lambda = 0.94$. The table in Fig. 10a shows an Event for $m = 12$. The local information possessed by R_0 is denoted by boldface characters in Fig. 10a (cf. the second paragraph of this section); the information hidden from R_0 is denoted as italic.

Based on the local information R_0 has, it classifies the indices $\alpha \in \{1, \dots, m\} \equiv \{a, b, \dots, l\}$ of the Event into the three classes defined above: 0011, XX10, XX0X. In the example of Fig. 10a, these classes can be identified as the corresponding check sets: 0011 = $\{b, e, f, i\}$, XX10 = $\{d, g\}$, and XX0X = $\{a, c, h, j, k, l\}$. This implies $m_{0011} \equiv \ell_1 = 4$, $m_{XX10} \equiv \ell_2 = 2$, and $m_{XX0X} \equiv \ell_3 = 6$. This local count list $\ell = (4, 2, 6)$ is in the domain $D(\zeta_R)$ of the incomplete strategy ζ_R ; in fact, it is in the orange region of Fig. 9. Hence R_0 can follow the incomplete strategy ζ_R , which results in the communication and decision steps depicted in Fig. 10b. There, the sender S and the receiver R_1 follow the Weak Broadcast protocol, and their output bit values are $x_S = 0$ and $y_1 = 1$, respectively, implying failure of the Weak Broadcast protocol.

The above example sheds light on why the strategy ζ_R in Eq. (20) is efficient (in fact, optimal in its domain) in achieving failure. In more general terms, its efficiency is reasoned as follows. According to this strategy, the adversary receiver R_0 sends a false data bit value ($y_{01} = 1 \neq 0 = x_S$) and tries to ‘back it up’ by a check set ρ_{01} that is convincingly long (that is, the Length Condition of the Cross-Check Phase is satisfied) and consistent (that is, to achieve that the Consistency Condition of the Cross-Check Phase is satisfied). This is achieved by the strategy ζ_R of Eq. (20), if the number of consistent indices is large enough; hence it is reasonable for R_0 to include *all* indices that are certainly consistent with the bit value $y_{01} = 1$ (red indices in Fig. 10), and include the minimal amount of indices which are only potentially consistent with the bit value $y_{01} = 1$ (these are the two blue indices in Fig. 10b), such that the Length Condition is satisfied.

C Failure probabilities

In this section, we compute the exact failure probability for the ‘no faulty’ adversary configuration, and compute tight upper bounds of the failure probabilities for the adversary configurations ‘ S faulty’ and ‘ R_0 faulty’. These results allow future users of the protocol to allocate resources sparingly. In fact, these results are analysed, and are used to optimize the protocol, in the main text.

C.1 No faulty

In this subsection, we consider the no faulty configuration. Without the loss of generality, assume that the data bit sent by S is $x_S = 0$. Then, we have the following exact result for the failure probability p_f , which we denote as $p_f^{(\text{nf})}$:

$$p_f^{(\text{nf})} = \sum_{m_{0011}=0}^{T-1} \binom{m}{m_{0011}} \left(\frac{1}{3}\right)^{m_{0011}} \left(\frac{2}{3}\right)^{m-m_{0011}}. \quad (24)$$

Recall the definition $T = \lceil \mu m \rceil$, introduced in the Weak Broadcast protocol, Sec. 2. Recall also that T is the minimal check set length that satisfies the Length Conditions.

To derive this result, it is sufficient to note that the protocol leads to failure only if the Length Condition of the Check Phase evaluates as False, that is, if the number of 0011 outcomes is less than the minimal check set length that satisfies the Length Conditions; i.e. if the following condition holds:

$$m_{0011} < T. \quad (25)$$

Since the Elementary Event 0011 is generated with a probability of $1/3$, the probability of Eq. (25) holding true can be expressed by the binomial formula Eq. (24). This concludes the proof. Note that the analysis generalises straightforwardly to the case when the message of the sender S is $x_S = 1$, and yields the same result for $p_f^{(\text{nf})}$.

C.2 S faulty

Here, we provide an upper bound $p_{f,\uparrow}^{(S)}$ for the failure probability in the S faulty configuration. To express this upper bound, recall that $T = \lceil \mu m \rceil$ is the minimal check set length that satisfies the Length Conditions, which was introduced in Eq. (13). Also recall that the integer Q was defined in Eq. (14). Here we use its exact definition, that is, $Q = T - \lceil T\lambda \rceil + 1$. With these notations, the upper bound $p_{f,\uparrow}^{(S)}$ reads

$$p_{f,\uparrow}^{(S)} = p_{f,\downarrow}^{(S)} + \left(1 - \sum_{\ell_3=T}^{m-T} \sum_{\ell_1=T-Q}^{m-Q-\ell_3} \binom{m}{\ell_3, \ell_1, m - \ell_1 - \ell_3} \left(\frac{1}{3}\right)^m\right), \quad (26)$$

with

$$p_{f,\downarrow}^{(S)} = \sum_{\ell_3=T}^{m-T} \sum_{\ell_1=T-Q}^{m-Q-\ell_3} \binom{m}{\ell_3, \ell_1, m - \ell_1 - \ell_3} \left(\frac{1}{3}\right)^m 2^{-Q}. \quad (27)$$

In the derivation of this upper bound, first we express the upper bound formally for a general incomplete strategy, and then we apply it to the S faulty strategy defined above.

Assume that we have identified an incomplete strategy ζ that is optimal in its domain $D(\zeta) \subset \text{LCL}_S$. An upper bound of the failure probability of this strategy can be expressed as

$$p_{f,\uparrow}^{(S)} = \sum_{\ell \in D(\zeta)} P_\zeta(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \bar{D}(\zeta)} P(\ell). \quad (28)$$

Here, $P(\ell)$ is the probability that a random Event produces the local count list ℓ . Furthermore, $P_\zeta(\text{failure}|\ell)$ is the conditional probability of failure, given the random Event produces the local count list ℓ . Finally, $\bar{D}(\zeta)$ is the complement of $D(\zeta)$, that is, $\bar{D}(\zeta) = \text{LCL}_S \setminus D(\zeta)$.

In words, the formula (28) for the upper bound is interpreted as follows: the random Event either produces a local count list ℓ that is in the domain of the strategy ζ , and then we compute the corresponding failure probability exactly (first sum); or, the Event produces a local count list ℓ that is outside the domain of the strategy ζ , and then we assume a worst-case scenario, i.e., that failure happens with unit probability (second sum).

Using the normalization condition

$$1 = \sum_{\ell \in \text{LCL}_S} P(\ell) = \sum_{\ell \in D(\zeta)} P(\ell) + \sum_{\ell \in \bar{D}(\zeta)} P(\ell), \quad (29)$$

we see that Eq. (28) can be reformulated as

$$p_{f,\uparrow}^{(S)} = \sum_{\ell \in D(\zeta)} P_\zeta(\text{failure}|\ell)P(\ell) + \left(1 - \sum_{\ell \in D(\zeta)} P(\ell)\right). \quad (30)$$

Note that these considerations suggest that for a given incomplete strategy, a lower bound $p_{f,\downarrow}^{(S)}$ of failure probability can also be expressed, by assuming a best-case scenario, i.e., zero failure probability, for Events that produce local count lists that are outside of the domain of the strategy. Formally, this lower bound is expressed as

$$p_{f,\downarrow}^{(S)} = \sum_{\ell \in D(\zeta)} P_{\zeta}(\text{failure}|\ell)P(\ell). \quad (31)$$

To evaluate the upper bound $p_{f,\uparrow}^{(S)}$ of the failure probability for this optimal partial strategy ζ via Eq. (30), we first express

$$P(\ell) = \binom{m}{\ell_1, m - \ell_1 - \ell_3, \ell_3} \left(\frac{1}{3}\right)^m. \quad (32)$$

Recall that $\ell = (\ell_1, \ell_2, \ell_3)$ is determined by two of its components, e.g., ℓ_1 and ℓ_3 , because $\ell_1 + \ell_2 + \ell_3 = m$. Equation (32) is a straightforward consequence of the fact that the probability of a 0011 outcome, the probability of a mixed outcome (0101 or 0110 or 1001 or 1010), and the probability of a 1100 outcome, are all equal to $1/3$. Finally, we express the failure probability of ζ on its domain, which is

$$P_{\zeta}(\text{failure}|\ell) = (1/2)^Q, \text{ for all } \ell \in D(\zeta). \quad (33)$$

This follows from the fact that S includes Q mixed indices in its check set σ_0 , and this leads to failure only if all the Q corresponding mixed outcomes contain a ‘1’ at R_0 . Inserting Eqs. (32) and (33) into Eq. (30), we obtain the result Eq. (27), which concludes our derivation. Note also that the first term of Eq. (27) is in fact the failure-probability lower bound of our specific strategy ζ , in line with the general expression in Eq. (31).

C.3 R_0 faulty

Here, we provide an upper bound $p_{f,\uparrow}^{(R)}$ for the failure probability in the R_0 faulty configuration. Again, recall that $T = \lceil \mu m \rceil$ (original definition in Eq. (13)) and $Q = T - \lceil T\lambda \rceil + 1$ (original definition in Eq. (14)). With these notations, the failure-probability upper bound reads:

$$p_{f,\uparrow}^{(R)} = p_{f,\downarrow}^{(R)} + \sum_{\ell_1=m-T+1}^m \binom{m}{\ell_1} \left(\frac{1}{3}\right)^{\ell_1} \left(\frac{2}{3}\right)^{m-\ell_1}, \quad (34)$$

where

$$\begin{aligned} p_{f,\downarrow}^{(R)} = & \sum_{\ell_1=T}^{m-T} \sum_{\ell_2=0}^{T-Q} \binom{m}{\ell_1, \ell_2, \ell_3} \left(\frac{1}{3}\right)^{\ell_1} \left(\frac{1}{6}\right)^{\ell_2} \left(\frac{1}{2}\right)^{\ell_3} \sum_{k=T-Q+1-\ell_2}^{T-\ell_2} \binom{T-\ell_2}{k} \left(\frac{2}{3}\right)^k \left(\frac{1}{3}\right)^{T-\ell_2-k} + \\ & + \sum_{\ell_1=T}^{m-T} \sum_{\ell_2=T-Q+1}^{m-\ell_1} \binom{m}{\ell_1, \ell_2, \ell_3} \left(\frac{1}{3}\right)^{\ell_1} \left(\frac{1}{6}\right)^{\ell_2} \left(\frac{1}{2}\right)^{\ell_3} + \\ & + \sum_{\ell_1=0}^{T-1} \binom{m}{\ell_1} \left(\frac{1}{3}\right)^{\ell_1} \left(\frac{2}{3}\right)^{m-\ell_1}. \end{aligned} \quad (35)$$

The result (34) is valid both for the $x_S = 0$ and the $x_S = 1$ scenario. In what follows, we derive this result. The derivation follows the 3-step scheme outlined in the S faulty case below Eq. (27).

Assume that we have identified an incomplete strategy ζ that is optimal on its domain $D(\zeta) \subset \text{LCL}_R$. An upper bound of the failure probability of this strategy can be expressed as shown in Eqs. (28). A lower bound can also be expressed as Eq. (31).

For our optimal incomplete strategy ζ , the formula for the failure-probability upper bound, Eq. (28), can be structured further, according to the three regions (orange, pink, blue) in Fig. 9:

$$p_{f,\uparrow}^{(R)} = \sum_{\ell \in \text{orange}} P_{\zeta}(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \text{pink}} P_{\zeta}(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \text{blue}} P_{\zeta}(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \bar{D}(\ell)} P(\ell). \quad (36)$$

A simplification is allowed by the observation that for local count lists (ℓ) in the pink and blue regions, the failure probability is $P_{\ell}(\text{failure}|\ell) = 1$, hence

$$p_{f,\uparrow}^{(R)} = \sum_{\ell \in \text{orange}} P_{\zeta}(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \text{pink}} P(\ell) + \sum_{\ell \in \text{blue}} P(\ell) + \sum_{\ell \in \bar{D}(\ell)} P(\ell). \quad (37)$$

In the pink region, failure is guaranteed, in fact, for any R_0 adversary strategy, since $\ell_1 < T$ implies that the check list $\sigma_0 = \sigma_1$ sent by S violates the Length Condition of the Check Phase. In the blue region, however, failure is guaranteed because the adversary R_0 was lucky to find enough outcomes XX10, that is, $\ell_2 \geq T - Q + 1$, such that its check set ρ_{01} automatically satisfies the Consistency Condition of the Cross-check Phase.

The last three terms of Eq. (37) can be readily expressed by binomial and trinomial distributions. The blue, pink, and green regions correspond to, respectively, to the second term of Eq. (35), to the third term of Eq. (35), and to the second term of Eq. (34).

To complete the derivation of the failure-probability upper bound, we express $P_\zeta(\text{failure}|\ell)$ for $\ell \in \text{orange}$:

$$P_\zeta(\text{failure}|\ell) = \sum_{k=T-Q+1-\ell_2}^{T-\ell_2} \binom{T-\ell_2}{k} \left(\frac{2}{3}\right)^k \left(\frac{1}{3}\right)^{T-\ell_2-k}, \text{ for } \ell \in \text{orange}. \quad (38)$$

To interpret this formula, let us first recall that in the orange region of $D(\zeta)$, it holds that $n_{\min} = T - \ell_2$. Furthermore, note also that the probability that the number of XX10 outcomes in the check set ρ_{01} of length $n_{\min} = T - \ell_2$ is k is given by the binomial distribution. Hence, Eq. (38) sums up the probabilities where k is large enough that ρ_{01} satisfies the Consistency Condition of the Cross-check Phase. Combining Eq. (38) with the trinomial distribution expressing the probability of the local count lists of the orange region, we obtain the first term of Eq. (35). This concludes the derivation of the failure-probability upper bound in the case of an adversary R_0 .

D Proofs of optimality

D.1 S faulty

Here, we show that the S faulty adversary strategy defined above is an optimal incomplete strategy, i.e., it is optimal on its domain.

Consider therefore an arbitrary strategy

$$\zeta_B = \left(k_{0011}^{(0)}, k_{\text{mixed}}^{(0)}, k_{1100}^{(0)}; k_{0011}^{(1)}, k_{\text{mixed}}^{(1)}, k_{1100}^{(1)} \right). \quad (39)$$

It is straightforward to see that the strategy

$$\zeta'_B = \left(k_{0011}^{(0)}, k_{\text{mixed}}^{(0)}, 0; 0, 0, m_{1100} \right), \quad (40)$$

derived from \mathcal{S}_B , has a higher or equal failure probability than the initial Bad Strategy \mathcal{S}_B : $p_f(\mathcal{S}_B) \leq p_f(\mathcal{S}'_B)$. The reason is as follows. On the one hand, deviating from $\left(k_{0011}^{(1)}, k_{\text{mixed}}^{(1)}, k_{1100}^{(1)} \right) = (0, 0, m_{1100})$ either decreases the failure probability, or does not change that. In words: from the viewpoint of an adversary S , it is useless to deviate from the correct communication toward R_1 . On the other hand, $k_{1100}^{(0)} > 0$ leads to weak broadcast, implying $p_f = 0$.

To analyze the failure probability of the ‘improved’ strategy ζ'_B further, we introduce $Q' = k_{\text{mixed}}^{(0)}$ and $T' = k_{0011}^{(0)} + Q'$. We address the question: are any strategies defined by $(T', Q') \neq (T, Q)$ better than the strategy the strategy ζ ? No, as we argue now.

(i) If $T' < T$, then the strategy ζ'_B is dysfunctional, $p_f = 0$, since there are not enough indices in σ_0 and σ_1 to convince the receivers; that is, the Length Condition of the Check Phase evaluates is violated, implying weak broadcast. These strategies are depicted as red circles in Fig. 11a.

(ii) If $T' \geq T$ and $Q' < Q$, then the strategy ζ'_B is dysfunctional, since this excludes the possibility that ρ_{01} contains enough inconsistent indices that compromise R_0 . These strategies are depicted as purple circles in Fig. 11.

(iii) The failure probability of the strategy ζ'_B along the ‘line’ $(T', Q') = (T + n, Q + n)$ decreases as $n \geq 0$ is increased. I.e., along this line, depicted as the line of diagonal arrows in Fig. 11a, the filled green circle depicting $n = 0$ represents the optimal strategy. The reason is that by increasing T' from T to $T + n$, the number of mixed outcomes in σ_0 for which R_0 needs to measure a ‘1’ to cause failure is increased to $Q + n$, as demanded by the Consistency Condition of the Cross-Check Phase. This reduces the corresponding failure probability from $(1/2)^Q$ to $(1/2)^{Q+n}$. This trend is depicted in Fig. 11a by the diagonal arrows.

(iv) The strategy ζ'_B is dysfunctional ‘below the line’ defined in (iii), see the blue circles in Fig. 11a. There, even if all Q' mixed outcomes included in σ_0 have ‘1’ at R_0 , that is insufficient to violate the Consistency Condition of the Cross-Check phase.

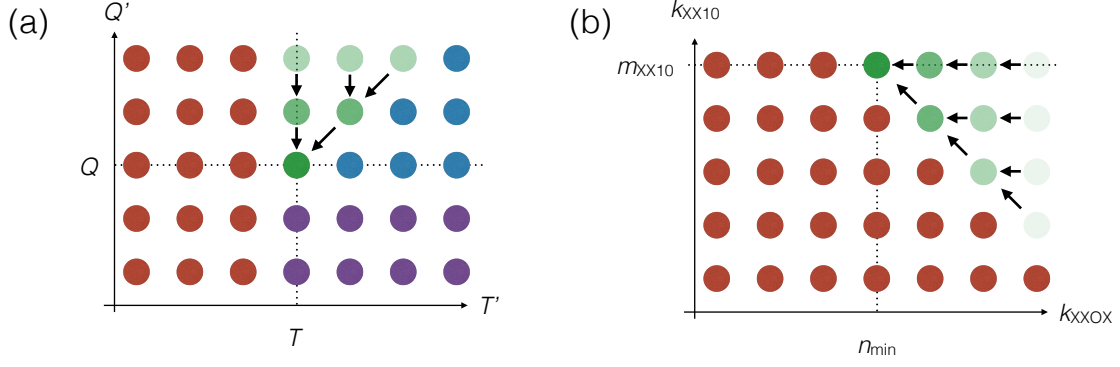


Figure 11: Optimal, suboptimal, and dysfunctional strategies. (a) Adversary strategies of a faulty sender S . (b) Adversary strategies of a faulty receiver R_0 . Filled green circle represents the incomplete optimal adversary strategy ζ_S in (a), ζ_R in (b). Red, purple, and blue circles represent dysfunctional strategies. Opaque green circles represent suboptimal adversary strategies. Arrows depict increasing failure probability.

(v) The failure probability of the strategies ζ'_B ‘above the line’ defined in (iii), that is, strategies with $(T', Q') = (T + n, Q + n')$ (where $n \geq 0$ and $n' \geq n$), decrease as n' is increased. Indeed, as n' is increased, the check set size remains unchanged ($T + n$), but the number of ‘guessed’ indices ($Q + n'$) is increased, and hence the failure probability $(1/2)^{Q+n'}$ decreases. This trend is depicted in Fig. 11a by the vertical arrows.

This concludes the proof that the strategy $\zeta = (T - Q, Q, 0; 0, 0, m_{1100})$ is indeed optimal on its domain.

D.2 R_0 faulty

Here, we prove that the R_0 adversary strategy $\zeta = (0, m_{XX10}, n_{\min})$ defined above is optimal on its domain.

(i) Increasing the first strategy parameter k_{0011} from zero decreases the failure probability, as it lowers the chance of the Consistency Condition of the Cross-check Phase evaluating as True.

(ii) Consider the alternative strategies $\zeta' = (0, k_{XX10}, k_{XX0X})$, with $k_{XX10} = m_{XX10} - n$ and $k_{XX0X} = n_{\min} + n'$, with $n, n' \geq 0$. The alternative strategies are dysfunctional ($p_f = 0$) ‘below the line’ of the (k_{XX10}, k_{XX0X}) plane defined by $n = n'$, i.e., for $n > n'$. This is because for such a strategy, the check set size $|\rho_{01}|$ is below the minimum size T required to satisfy the Length Conditions. These dysfunctional strategies are shown as the red circles in Fig. 11b.

(iii) The alternative strategies along the line $n = n'$, shown in Fig. 11b as the line of diagonal arrows, have a decreasing failure probability as n is increased. Along this line, the check set size $|\rho_{01}|$ is fixed to the minimum size T , but the number $n_{\min} + n'$ of guessed indices increases, and hence the probability of satisfying the Consistency Condition of the Cross-check Phase decreases. This trend is depicted in Fig. 11b by the diagonal arrows.

(iv) Consider the alternative strategies ‘above the line’ of the (k_{XX10}, k_{XX0X}) plane defined by $n = n'$, i.e., above the line of diagonal arrows in Fig. 11b. These are the strategies of the form $\zeta = (0, m_{XX10} - n, n_{\min} + n')$ with $n' \geq n$. For a fixed n , the failure probability of these strategies decreases as n' is increased. This is because increasing n' by 1 increases the check set size $|\rho_{01}|$ by 1, and hence the right hand side of the Consistency Condition of the Check Phase by 1, and at the same time, increases the minimum number of lucky indices needed to satisfy that Consistency Condition, leading to a decrease of the failure probability. This trend is depicted in Fig. 11b by the horizontal arrows.

This concludes the proof that the strategy $\zeta = (0, m_{XX10}, n_{\min})$ is indeed optimal on its domain

E Security proof

In this Appendix, we derive asymptotic ($m \rightarrow \infty$) upper bounds for the failure probability for generic adversary strategies.

E.1 No faulty

Without the loss of generality, we assume that S sends the message $x_S = 0$.

The protocol will achieve consensus, $y_0 = y_1 = x_S = 0$, if the number of 0011 outcomes in the event is at least T , cf. the exact failure-probability result of Eq. (24). The probability of achieving consensus, i.e., that a random event contains the outcome 0011 with frequency m_{0011} is

$$P(m_{0011}) = P\left(\left(\sum_{j=1}^m X_j\right) = m_{0011}\right), \quad (41)$$

where X_j are binary (Bernoulli) random variables, taking values 0 or 1, with probability $p = 1/3$ for the latter. This implies that

$$p_f(m) = P\left(\left(\sum_{j=1}^m X_j\right) \leq \mu m\right) \quad (42)$$

Recall that $\mu < 1/3$; hence, the failure probability p_f can be upper bounded by the Chernoff bound as

$$p_f(m) \leq \exp\left(-\frac{1}{2}\frac{m}{3}(1-3\mu)^2\right). \quad (43)$$

That is, in the absence of an adversary, the failure probability of the Weak Broadcast protocol is subject to an upper bound that decreases exponentially as m increases.

Recall that the general form of the Chernoff bound used to derive this result is

$$P(X \leq (1-\delta)\bar{X}) \leq e^{-\bar{X}\delta^2/2}, \text{ for } 0 \leq \delta, \quad (44)$$

where \bar{X} is the expectation value of the sum variable X .

E.2 S faulty

The security proof for the S faulty adversary configuration will be based on the tight failure-probability upper bound calculation presented in App. C.2.

Our goal is here is to derive an exponentially decaying upper bound for the failure probability. To this end, we restrict the domain of the S faulty adversary strategy ζ , defined in C.2. The restricted strategy is denoted as ζ_r . In particular, we restrict the domain of ζ for ‘regular’ or ‘typical’ local count lists, where ℓ_1 and ℓ_3 are approximately equal to their expected values $m/3$. Formally, we define $D(\zeta_r)$ as formed by those (ℓ_1, ℓ_3) pairs where

$$\mu m < \ell_1, \ell_3 < 2m/3 - \mu m. \quad (45)$$

When is ζ_r indeed a restriction of ζ ? Conditions (15) and (17) are automatically satisfied if Eq. (45) holds. However, Eq. (16) demands $Q \leq \ell_2$; this is also guaranteed if $\lambda \geq 1/2$.

Since $D(\zeta_r) \subset D(\zeta)$, and $D(\zeta)$ is optimal on its domain, the restricted strategy ζ_r is also optimal on its domain.

In the spirit of Eq. (28), the failure-probability upper bound can be expressed using the restricted strategy ζ_r as follows:

$$p_f(m) \leq \sum_{\ell \in D(\zeta_r)} P_{\zeta_r}(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \bar{D}(\zeta_r)} P(\ell). \quad (46)$$

It is straightforward to exponentially upper bound the first term using the fact that

$$P_{\zeta_r}(\text{failure}|\ell) = 2^{-Q} \approx 2^{-(1-\lambda)\mu m}, \quad (47)$$

which is independent of ℓ . Therefore, we have

$$\sum_{\ell \in D(\zeta_r)} P_{\zeta_r}(\text{failure}|\ell)P(\ell) \approx 2^{-(1-\lambda)\mu m} \sum_{\ell \in D(\zeta_r)} P(\ell) \leq 2^{-(1-\lambda)\mu m}, \quad (48)$$

where we used $\sum_{\ell \in D(\zeta_r)} P(\ell) \leq 1$.

To exponentially upper bound the second term of Eq. (46), we express it as

$$\begin{aligned} \sum_{\ell \in \bar{D}(\zeta_r)} P(\ell) &= P((\ell_1 \leq \mu m) \text{ OR } (\ell_1 \geq 2m/3 - \mu m) \text{ OR } (\ell_3 \leq \mu m/3) \text{ OR } (\ell_3 \geq 2m/3 - \mu m)) \\ &= P(\ell_1 \leq \mu m) + P(\ell_1 \geq 2m/3 - \mu m) + P(\ell_3 \leq \mu m/3) + P(\ell_3 \geq 2m/3 - \mu m) \end{aligned} \quad (49)$$

$$\leq 4e^{-\frac{1}{3}\frac{m}{3}(1-3\mu)^2}. \quad (50)$$

In the last step, we used another Chernoff bound (cf. Eq. (44)):

$$P(|X - \bar{X}| \geq \delta \cdot \bar{X}) \leq 2e^{-\bar{X}\delta^2/3}, \text{ for } 0 \leq \delta. \quad (51)$$

E.3 R_0 faulty

Here, we describe the security proof for the R_0 faulty adversary configuration. This proof is based on the tight failure-probability upper bound calculation presented in App. C.3.

We restrict the domain of the R_0 faulty optimal incomplete adversary strategy ζ , defined in B.2. The restricted strategy is denoted as ζ_r . In particular, we restrict the domain of ζ for ‘regular’ local count lists, where ℓ_1 and ℓ_2 are close to their expected values, $m/3$ and $m/6$, respectively. Formally, we define the domain $D(\zeta_r)$ of the restricted strategy as formed by those (ℓ_1, ℓ_2) pairs which satisfy

$$\mu m < \ell_1 < 2m/3 - \mu m, \quad (52)$$

$$\mu m/2 < \ell_2 < 2m/6 - \mu m/2. \quad (53)$$

Note that ζ_r is indeed a restriction of ζ , since the upper bound in Eq. (52) is a stricter condition than (23). Furthermore, since $D(\zeta_r) \subset D(\zeta)$, and ζ is optimal on its domain, the restricted strategy ζ_r is also optimal on its domain.

In the spirit of Eq. (28), the failure-probability upper bound can be expressed using the restricted strategy ζ_r as follows:

$$p_f(m) \leq \sum_{\ell \in D(\zeta_r)} P_{\zeta_r}(\text{failure}|\ell)P(\ell) + \sum_{\ell \in \bar{D}(\zeta_r)} P(\ell). \quad (54)$$

To exponentially upper bound the second term of Eq. (54), we express it as

$$\begin{aligned} \sum_{\ell \in \bar{D}(\zeta_r)} P(\ell) &= P((\ell_1 \leq \mu m) \text{ OR } (\ell_1 \geq 2m/3 - \mu m) \text{ OR } (\ell_2 \leq \mu m/6) \text{ OR } (\ell_2 \geq 2m/6 - \mu m/2)) \\ &= P(\ell_1 \leq \mu m) + P(\ell_1 \geq 2m/3 - \mu m) + P(\ell_2 \leq \mu m/6) + P(\ell_2 \geq 2m/6 - \mu m/2) \end{aligned} \quad (55)$$

$$\leq 2e^{-\frac{1}{3}\frac{m}{3}(1-3\mu)^2} + 2e^{-\frac{1}{3}\frac{m}{6}(1-3\mu)^2}. \quad (56)$$

In the last step, we used the Chernoff bound in Eq. (51).

Finally, we provide an upper bound for the first term of Eq. (54). Within the restricted strategy domain $D(\zeta_r)$, the adversary R_0 has the greatest chance to cause failure if the number of ‘safe’ indices, ℓ_2 , is maximal, $\ell_2 = \ell_{2,\max} \equiv \lfloor 2m/6 - \mu m/2 \rfloor$. For a local count list containing $\ell_{2,\max}$, Eq. (38) specifies $P_{\zeta}(\text{failure}|\ell_1, \ell_{2,\max})$ as a sum of the upper part of a binomial distribution, between $k = T - Q + 1 - \ell_{2,\max}$ to $k = T - \ell_{2,\max}$. In fact, the quantity $P_{\zeta_r}(\text{failure}|\ell_1, \ell_{2,\max})$ is independent of ℓ_1 .

The Chernoff bound

$$P(X \geq (1 + \delta)\bar{X}) \leq e^{-\bar{X}\delta^2/3}, \text{ for } 0 \leq \delta, \quad (57)$$

can be used to upper bound this sum, if the latter is an upper tail sum, that is, the lower end of the k sum is above the expectation value $\bar{X} = \frac{2}{3}(T - \ell_{2,\max})$ of k :

$$T - Q + 1 - \ell_{2,\max} > \frac{2}{3}(T - \ell_{2,\max}). \quad (58)$$

The latter condition is readily transformed as

$$\lambda > \frac{2 + 9\mu}{18\mu}, \quad (59)$$

where we disregarded ceiling and floor functions for simplicity. Note that since $\lambda < 1$, the condition of Eq. (59) can be satisfied only if

$$\mu > 2/9 \approx 0.222. \quad (60)$$

If the condition (59) holds, then the Chernoff bound of Eq. (57) is applied with the substitution

$$\delta = \frac{2 + 9\mu - 18\lambda\mu}{6 - 27\mu}, \quad (61)$$

yielding

$$\sum_{\ell \in D(\zeta_r)} P_{\zeta_r}(\text{failure}|\ell)P(\ell) \leq \sum_{\ell \in D(\zeta_r)} P_{\zeta_r}(\text{failure}|\ell_1, \ell_{2,\max})P(\ell) \quad (62)$$

$$= P_{\zeta_r}(\text{failure}|\ell_{2,\max}) \sum_{\ell \in D(\zeta_r)} P(\ell) \quad (63)$$

$$\leq P_{\zeta_r}(\text{failure}|\ell_{2,\max}) \quad (64)$$

$$\leq e^{-\bar{X}\delta^2/3}, \quad (65)$$

with

$$\bar{X} = \frac{2}{3}(T - \ell_{2,\max}) \approx \left(\frac{3}{2}\mu - \frac{1}{3}\right)m. \quad (66)$$

Note that Eq. (60) implies that the coefficient of m on the right hand side is positive, i.e., the upper bound in Eq. (66) is exponentially decreasing with increasing m .

F Preparation and benchmarking of the four-qubit singlet state on IBM Q

This appendix exhibits the data we obtained by conducting experiments on IBM Q quantum computer prototypes. We aimed these experiments to prepare the four-qubit singlet state $|\psi\rangle$ of Eq. (1) using Circuit A of Fig. 5, and to quantify how close the experiment resembles the preparation of an ideal four-qubit singlet state. Further details are in the table caption.

backend	qubits	F_c	F_c^{mitig}	date(c)	F_q	F_q^{mitig}	date(q)
ibmq_casablanca	0-1-3-5	0.3234	0.3115	2021.04.26	0.1812	0.2071	2021.04.28
	2-1-3-5	0.3109	0.3033	2021.04.26	0.1963	0.2166	2021.04.28
	1-3-5-4	0.3409	0.1259	2021.04.26	0.3182	0.3436	2021.04.28
	1-3-5-6	0.3066	0.1182	2021.04.26	0.1118	0.1105	2021.04.28
	5-3-1-0	0.3061	0.1566	2021.04.26	0.5006	0.5527	2021.04.28
	5-3-1-2	0.3130	0.1587	2021.04.26	0.3726	0.4141	2021.04.28
	4-5-3-1	0.3399	0.3011	2021.04.26	0.1701	0.1818	2021.04.28
6-5-3-1	0.3532	0.2047	2021.04.26	0.1301	0.1324	2021.04.28	
ibmq_santiago	0-1-2-3	0.8510	0.9021	2021.04.25	0.7435	0.8116	2021.04.26
	1-2-3-4	0.8501	0.9002	2021.04.25	0.7221	0.8011	2021.04.26
	3-2-1-0	0.7595	0.8604	2021.04.25	0.5431	0.5939	2021.04.26
	4-3-2-1	0.8332	0.8795	2021.04.25	0.6659	0.7803	2021.04.26
ibmq_athens	0-1-2-3	0.7836	0.8370	2021.04.23	0.6632	0.7311	2021.04.26
	1-2-3-4	0.8050	0.8608	2021.04.23	0.6652	0.7868	2021.04.26
	3-2-1-0	0.7686	0.8828	2021.04.23	0.5599	0.6283	2021.04.26
4-3-2-1	0.8025	0.8573	2021.04.23	0.6847	0.7811	2021.04.26	
ibmq_bogota	0-1-2-3	0.7566	0.8840	2021.04.26	0.4710	0.6022	2021.04.26
	1-2-3-4	0.7322	0.8813	2021.04.26	0.5278	0.7802	2021.04.27
	3-2-1-0	0.6904	0.7909	2021.04.26	0.4998	0.6512	2021.04.27
4-3-2-1	0.7295	0.8352	2021.04.26	0.4003	0.6269	2021.04.27	
ibmq_rome	0-1-2-3	0.6347	0.7103	2021.04.26	0.3893	0.4727	2021.04.26
	1-2-3-4	0.7106	0.7896	2021.04.26	0.5587	0.6827	2021.04.26
	3-2-1-0	0.5531	0.6039	2021.04.26	0.1537	0.1734	2021.04.26
	4-3-2-1	0.6424	0.7083	2021.04.26	0.3775	0.4475	2021.04.26

Table 3: Preparation and benchmarking of the four-qubit singlet state on IBM Q prototype quantum computers. We prepared the state $|\psi\rangle$ of Eq. (1) using Circuit A shown in Fig. 5. The first column indicates the quantum computer ('backend') used; the second column identifies the qubits of Circuit A with the qubits of the backend. The third (fourth) column, F_c (F_c^{mitig}), shows the classical fidelity, Eq. (4), of the ideal distribution (Fig. 1c) and the distribution generated by the backend, without (with) readout error mitigation. The third (fourth) column, F_c (F_c^{mitig}), shows the classical fidelity, Eq. (4), of the ideal distribution (Fig. 1c) and the distribution generated by the backend, without (with) readout error mitigation. The sixth (seventh) column, F_q (F_q^{mitig}), shows the quantum state fidelity, Eq. (5), of $|\psi\rangle$ and the output density matrix of Circuit A, as inferred from quantum state tomography without (with) readout error mitigation. Measurement dates are indicated in the fourth and eighth columns.