Optimal Communication Complexity of Authenticated Byzantine Agreement

Atsuki Momose
Nagoya University, Aichi, Japan
Intelligent Systems Laboratory, SECOM CO.,LTD., Tokyo, Japan

Ling Ren
University of Illinois at Urbana-Champaign, Urbana, IL, USA

Abstract

Byzantine Agreement (BA) is one of the most fundamental problems in distributed computing, and its communication complexity is an important efficiency metric. It is well known that quadratic communication is necessary for BA in the worst case due to a lower bound by Dolev and Reischuk. This lower bound has been shown to be tight for the unauthenticated setting with $f < n/3$ by Berman et al. but a considerable gap remains for the authenticated setting with $n/3 \leq f < n/2$.

This paper provides two results towards closing this gap. Both protocols have a quadratic communication complexity and have different trade-offs in resilience and assumptions. The first protocol achieves the optimal resilience of $f < n/2$ but requires a trusted setup for threshold signature. The second protocol achieves near optimal resilience $f \leq (1/2 - \varepsilon)n$ in the standard PKI model.

2012 ACM Subject Classification Security and privacy → Distributed systems security

Keywords and phrases Byzantine Agreement, Communication Complexity, Lower Bound

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.32

Acknowledgements We would like to thank Zhuolun Xiang for helpful feedback.

1 Introduction

Byzantine Agreement (BA) is one of the most fundamental problems in distributed algorithms [21]. It also serves as an important building block in cryptography and distributed systems. At a high level, Byzantine agreement is the problem for n parties to agree on a value, despite that up to f of them may behave arbitrarily (called Byzantine faults). Arguably the most important efficiency metric of Byzantine Agreement is the communication complexity, since communication will be the bottleneck in applications like state machine replication and cryptocurrency when there is a large number of parties.

Dolev and Reischuk proved that a quadratic number of messages are necessary for any perfectly secure BA protocol. More formally, they showed that even in the authenticated setting (i.e., assuming public key infrastructure and ideal digital signature), any BA protocol with perfect security (i.e., all executions are correct) has at least one execution where quadratic number of messages are sent by honest parties. The tightness of this lower bound was partially established by Berman et al. in the unauthenticated setting with $f < n/3$. However, for decades, the best known protocol for the authenticated setting (with $f \geq n/3$) remains the classic Dolev-Strong protocol [14] [1], which uses quadratic messages but cubic communication. The reason is that in Dolev-Strong, the messages can contain up to $f + 1$ signatures. Therefore, the optimal worst-case communication complexity of authenticated BA with $f \geq n/3$ has remained an open problem for decades.

[1] Dolev-Strong solves a related problem called Byzantine broadcast, but it is easy to transform it into a BA protocol.

Table 1 Upper bounds for worst-case communication complexity of Byzantine agreement with assumptions preserving the quadratic lower bound. ε is any positive constant.

protocol	model	communication	resilience
Berman et al. [6]	unauthenticated	$O(n^2)$	$f < n/3$
Dolev-Strong [14]	authenticated	$O(\kappa n^2 + n^3)$ [a]	$f < n/2$ [b]
this paper	threshold signature	$O(\kappa n^2)$	$f < n/2$
this paper	authenticated	$O(\kappa n^2)$	$f \leq (\frac{1}{2} - \varepsilon)n$

a) The original Dolev-Strong protocol solves BB but can be easily converted into a BA protocol with an initial round to multicast the inputs. Using a multi-signature with a list of signer identities attached, the protocol achieves $O(\kappa n^2 + n^3)$.

b) Although the original Dolev-Strong BB protocol tolerates $f < n$ faults, converting it to a BA protocol decreases the fault tolerance to $f < n/2$, which is optimal for authenticated BA.

This paper provides two results that help close this gap. More specifically, we show the following two theorems. Note that when $f \geq n/3$, it is necessary to adopt the synchronous and authenticated setting. Under asynchrony [18], partial synchrony [15], or the unauthenticated setting [17], BA is impossible for $f \geq n/3$.

▶ **Theorem 1.** *Assuming a threshold signature scheme, there exists a Byzantine agreement protocol with $O(\kappa n^2)$ communication complexity tolerating $f < n/2$ faults where n is the number of parties and κ is a security parameter.*

▶ **Theorem 2.** *Assuming a digital signature scheme with a public-key infrastructure, there exists a Byzantine agreement protocol with $O(\kappa n^2)$ communication complexity tolerating $f \leq (\frac{1}{2} - \varepsilon)n$ faults where n is the number of parties, κ is a security parameter, and ε is any positive constant.*

As we can see, the above two results achieve quadratic worst-case communication with different trade-offs. The first result achieves the optimal resilience $f < n/2$ but relies on a trusted setup due to the use of threshold signature. On the other hand, the second result is in the standard PKI model, but there is a small gap in the resilience.

Tightness with respect to the Dolev-Reischuk lower bound. In the Byzantine agreement and Byzantine fault tolerance literature, it is common and convenient to abstract signatures as ideal oracles and focus on the aspect of distributed computing [21, 14, 13, 9]. The rational is that modern cryptography has given us solid understandings and confidence about digital signatures, and that the probability that an adversary breaks a signature scheme is too small to be a concern.

When we abstract digital signatures and threshold signatures as ideal oracles with perfect security, our two results match the quadratic worst-case communication lower bound established by Dolev and Reischuk. Table 1 compares our results to the current landscape of worst-case communication complexity of perfectly secure BA.

On this note, it is very important to note that the Dolev-Reischuk lower bound applies to any protocol that is perfectly secure, even if the protocol has access to ideal digital signature and threshold signature oracles. With ideal digital signature and threshold signature oracles, our protocols are perfectly secure. On the other hand, there exist in the literature sub-quadratic BA protocols [20, 10, 1, 11] that use randomization techniques and allow a negligible

fraction of the executions to fail. These protocols do not provide perfect security even if we assume their randomization primitives are ideal. Naturally, they do not address the tightness of the Dolev-Reischuk lower bound.

We also remark that while it is possible to circumvent the Dolev-Reischuk lower bound by allowing a small failure probability, it should be clear that if the only source of failure in a protocol comes from imperfect cryptographic primitive, the protocol will not be able to circumvent the lower bound, because upgrading the cryptographic primitive from imperfect security to perfect security (at no extra costs) only strengthens the protocol.

Comparing with state-of-the-art BA solutions. Although our primary motivation of this study is to show the tightness of the quadratic lower bound of Dolev-Reischuk, the second result in Theorem 2 has some advantage even over state-of-the-art BA protocols with assumptions that are not subject to the Dolev-Reischuk bound. To the best of our knowledge, our second protocol is the first to achieve the following three properties simultaneously under the standard PKI model: (1) near-optimal resilience of $f \leq (\frac{1}{2} - \varepsilon)n$, (2) security against an adaptive adversary, (3) expected sub-cubic communication complexity. In fact, our protocol achieves worst-case quadratic communication and is secure against a strongly rushing (defined in [1]) adaptive adversary. The works of Berman et al. [6] and King-Saia [20] achieve (sub-)quadratic communication and adaptive security but tolerate only $f < n/3$. Abraham et al. [2, 1] achieve (sub-)quadratic communication and adaptive security under $f \leq (\frac{1}{2} - \varepsilon)n$, but require some trusted setup assumption due to the use of threshold signature or verifiale random functions. Tsimos et al. [28] recently achieve nearly-quadratic communication in the standard PKI model for $f \leq (1 - \varepsilon)n$ (for broadcast), but it is secure only against a static adversary.

Organization. The rest of the paper is organized as follows. In the rest of this section, we briefly review related work and give an overview of the techniques we use to achieve our two results. Section 2 introduces definitions, models and notations. Section 3 introduces the recursive framework to get a BA protocol with quadratic communication including the definition of GBA primitive. Section 4 presents two GBA protocols to instantiate two BA protocols with different trade-offs to complete our results. Finally, we discuss future directions and conclude the paper in Section 5.

1.1 Technical Overview

Abstracting the recursive framework of Berman et al. To obtain the results, we revisit the Berman et al. [6] protocol. At a high level, Berman et al. is a recursive protocol: it partitions parties into two halves recursively until they reach a small instance with sufficiently few (e.g., a constant number of) participants. Since the upper bound on the fraction of faults 1/3 is preserved in at least one of two halves, the "correct" half directs the entire parties to reach an agreement. If the communication except the two recursive calls is quadratic, the communication complexity of the entire protocol is also quadratic. The challenge is to prevent an "incorrect" run of recursive call (in a half with more than 1/3 faults) from ruining the result. Berman et al. solve this problem with a few additional rounds of communication called "universal exchange" before each recursive call. It helps honest parties stick to a value when all honest parties already agree on the value, thus preventing an incorrect recursive call from changing the agreed-upon value.

Back to our setting of $f \geq n/3$, we will use the recursive framework of Berman et al.. However, the universal exchange step of Berman et al. relies on a quorum-intersection argument, which only works under $f < n/3$. To elaborate, the quorum size can be at most $n-f$; two quorums of size $n-f$ intersect at $2(n-f)-n = n-2f$ parties; for this intersection to contain at least one honest party, it requires $n-2f > f$, or equivalently $f < n/3$.

To achieve our goal of BA with $f \geq n/3$, we observe that the functionality achieved by the universal exchange can be abstracted as a primitive called graded Byzantine agreement (GBA), which we formally define in Section 3. If we can construct a GBA with quadratic communication and plug it into the recursive framework, we will obtain a BA protocol with quadratic communication. Thus, it remains to construct quadratic GBA.

Two constructions of Graded BA with different trade-offs. As the name suggests, the GBA primitive shares some similarities with graded broadcast studied in [16, 19, 2], but it is harder to construct due to the fact that every party has an input. This can be addressed in two ways, leading to our two constructions.

The first method way is to resort to the (well-established) use of threshold signatures [7, 29]. Roughly, a threshold signature condenses a quorum of $n-f = \Omega(n)$ votes into a succinct proof of the voting result. This way, a verifiable voting result can be multicasted to all parties using quadratic total communication (linear per node). This achieves Theorem 1 and requires a trusted setup for threshold signature.

Next, we try to construct a quadratic GBA without trusted setup or threshold signature scheme. This turns out to be much more challenging. Naïvely multicasting the voting result would require quadratic communication per node (cubic in total) since the voting result consists of a linear number of votes. To get around this problem, we replace the multicast step with communication through an expander graph with constant degree. As each party transmits the voting result to only a constant number of neighbors, the communication is kept quadratic in total even though the voting result consists of a linear number of votes. Our key observation is that even though some of the honest parties may fail to receive or transmit the voting result (because all their neighbors are corrupted), as long as a small but linear number of honest parties transmit the voting result, the good connectivity of the expander helps prevent inconsistent decisions between honest parties. In order to verify a linear number of honest parties actually transmit, a quorum of $n-f$ parties who claim to have transmitted should contain at least a linear number of honest parties, which results in the gap of ϵn in the resilience in Theorem 2.

1.2 Related Work

Byzantine Agreement was first introduced by Lamport et al. [26, 21]. Without cryptography (i.e., the unauthenticated setting), BA can be solved if and only if $f < n/3$. Assuming a digital signature scheme with a public-key infrastructure (i.e., the authenticated setting), BA can be solved if and only if $f < n/2$. Lamport et al. gave BA protocols for both settings, but they both require exponential communication. Later, polynomial communication protocols were shown in both settings. In particular, Dolev and Strong [14] showed a $O(\kappa n^3)$ communication protocol for the authenticated setting and Dolev et al. [12] showed a $O(n^3 \log n)$ communication protocol for the unauthenticated setting. For the unauthenticated setting, Berman et al. further reduced the communication to $O(n^2)$, matching a lower bound established by Dolev and Reischuk [13], which states that any deterministic protocol with perfect security even in the authenticated setting must incur $\Omega(n^2)$ communication complexity. A recent work called HotStuff [29] can be modified [27] to achieve $O(\kappa n^2)$ communication with $f < n/3$ for the authenticated setting.

We also mention several orthogonal lines of work. Some works known as extension protocols [8, 24, 25, 22] achieve an optimal $O(nl)$ communication complexity for sufficiently long inputs of size l using the BA oracle for short inputs. When the input size is small, e.g., $l = O(1)$, the communication complexity degenerates to that of the underlying BA oracle. Our work provides improved oracles for these protocols.

Another line of works study protocols with sub-quadratic communication [20, 10, 1]. The idea is to select a random and unpredictable subset of parties to run the protocol (often using cryptographic primitives such as verifiable random function). Even after assuming ideal common randomness, these protocols will still have a small fraction of insecure executions, and are thus not subject to the Dolev-Reischuk lower bound. In contrast, we only use (threshold) signatures for message authentication. Once we assume ideal (threshold) signatures, our protocols are perfectly secure and are subject to the Dolev-Reischuk.

Other works study protocols with expected quadratic communication protocols [16, 19, 7, 23, 4, 2]. These protocols can require super-quadratic communication in the worst-case.

2 Preliminaries

Execution model. We define a protocol as an algorithm for a set of parties. There are a set of n parties, of which at most $f < n$ are Byzantine faulty and behave arbitrarily. We assume $f = \Theta(n)$. All presented protocols are secure against f adaptive corruption that can happen anytime during the protocol execution. Moreover, we assume a strongly rushing adaptive adversary [2, 1] who can corrupt parties in a round after seeing the messages they sent in that round and immediately delete those messages from network before they reach other parties. A party that is not faulty throughout the execution is said to be honest and faithfully execute the protocol. We use the term *quorum* to mean the minimum number of all honest parties, i.e., $n - f$. A protocol proceeds in synchronous rounds. If an honest party sends a message at the beginning of some round, an honest recipient receives the message at the end of that round.

Ideal (threshold) signatures. As mentioned, our two results, after assuming an ideal signatures and threshold signatures, address the tightness of the Dolev-Resichuk lower bound. We define the interface of signature and threshold signature oracles.

▶ **Definition 3** (Digital signature). *A digital signature oracle provides the following interfaces:*

- $\sigma \leftarrow \mathsf{Sign}_r(x)$. *Party r can invoke this interface to obtain a signature σ by party r on message x.*
- $b \leftarrow \mathsf{Verify}(\sigma, x, r)$. *Any party can invoke this interface to check whether σ is a signature by party r on message x.*

The oracle satisfies the following property.

- *For any σ, x, r, $\mathsf{Verify}(\sigma, x, r)$ outputs $b = 1$ if and only if $\mathsf{Sign}_r(x)$ has been queried by party r and the output is σ.*

The above property ensures *correctness*, i.e., correctly generated signatures are always verified, and *unforgeability*, i.e., no one other than party r can generate a signature for party r. For simplicity, we use $\langle x \rangle_r$ to denote a signed message x by party r, i.e., $\langle x \rangle_r = (x, \sigma)$ where $\sigma = \mathsf{Sign}_r(x)$ Any party can verify a signed message $\langle x \rangle_r = (x, \sigma)$ by querying $\mathsf{Verify}(\sigma, x, r)$.

▶ **Definition 4** ((t, n)-threshold signature)**.** *Each party r have access to the (t, n)-threshold signature oracle that provides the following interfaces, where $t < n/2$ is a given threshold.*

- $\sigma \leftarrow \mathsf{Sign}_r(x)$. *Party r can invoke this interface to obtain a signature share σ by party r on message x.*
- $b \leftarrow \mathsf{VerifyShare}(\sigma, x, r)$. *Any party can invoke this interface to check whether σ is a signature share by party r on message x.*
- $\Sigma \leftarrow \mathsf{Combine}(x, \{\sigma_1, ..\sigma_t\}, \{r_1, ..r_t\})$. *Any party can invoke this interface to combine a set of signature shares $\{\sigma_1, ..\sigma_t\}$ on the message x from t different parties $\{r_1, ..r_t\}$ into a threshold signature Σ.*
- $b \leftarrow \mathsf{Verify}(x, \Sigma)$. *Any party can invoke this interface to check whether Σ is a threshold signature generated from valid t signature shares.*

The oracle satisfies the following properties.

- *For any σ, x, r, $\mathsf{VerifyShare}(\sigma, x, r)$ outputs $b = 1$ if and only if $\mathsf{Sign}_r(x)$ has been queried by party r and the output is σ.*
- *For any x, $\mathsf{Verify}(x, \Sigma)$ outputs $b = 1$ if and only if there exist $\{\sigma_1, ..\sigma_t\}$ and $\{r_1, ..r_t\}$ such that for all $1 \leq i \leq t$, $\mathsf{VerifyShare}(\sigma_i, x, r_i) = 1$, and $\mathsf{Combine}(x, \{\sigma_1, ..\sigma_t\}, \{r_1, ..r_t\})$ has been queried by a party and the output is Σ.*

These two properties together satisfy the correctness and unforgeability properties of the signature shares and threshold signatures as before, and in addition a *robustness* property, i.e., t valid signature shares can always be combined into a valid threshold signature.

For simplicity, we use the same notation $\langle x \rangle_r$ as in digital signature to denote a tuple of message x and a signature share $\sigma \leftarrow \mathsf{Sign}_r(x)$. Each party r verifies a signature share $\langle x \rangle_r = (x, \sigma)$ by querying $\mathsf{VerifyShare}(\sigma, x, r)$. A set of $\langle x \rangle_*$ from t different parties can be combined into a threshold-signed x and verified by any party, using the $\mathsf{Combine}$ and Verify interfaces.

Setup assumptions. In practice, the above oracles are realized with negligible error with a PKI setup or trusted setup. The currently known threshold signature schemes require a trusted dealer who generates all public and private keys for all parties and a group public key to verify a combined full signature, henceforth we call it trusted setup. The digital signature requires the standard PKI setup and does not require any trusted setup beyond that. In that case, each party independently generates a pair of public and private keys without any extra assumption.

The Dolev-Reischuk lower bound. The Dolev-Reischuk lower bound holds (without any modification) even with ideal (threshold) signature oracles. We also note that the Dolev-Reischuk lower bound, which was originally proved for deterministic protocols, can be extended to randomized protocols as well. More precisely, any BA protocol (either deterministic or randomized) cannot simultaneously enjoy perfect security and sub-quadratic worst-case communication complexity. This has been observed and briefly mentioned in [20] and we show a proof for completeness.

▶ **Theorem 5.** *There does not exist a (either deterministic or randomized) BA protocol with worst-case communication complexity of at most $f^2/4$ that is perfectly secure.*

Proof. Suppose for the sake of contradiction that there exists such a protocol P. If P is randomized, we can transform P into a deterministic protocol P^* by fixing the output of the all random coin tossing to 0. Since P if perfectly secure and has at most $f^2/4$ communication

cost in the worst case, P^* is a deterministic BA protocol that is perfectly secure and has at most $f^2/4$ communication complexity. This contradicts the original Dolev-Reischuk lower bound [13]. ◀

Give this more general lower bound, our protocols, regardless of whether or not ideal digital signatures and threshold signatures are considered deterministic or randomized, are subject to the quadratic worst-case lower bound.

A remark on complexity metrics. The communication complexity of a protocol is the maximum number of bits sent by all honest parties combined across all executions. Since all messages in our protocols are signed, we use the signature size κ as the unit of measure for communication. We assume the size of any input value is on the order of κ. The Dolev-Reichuk lower bound, however, is in terms of the number of messages. With no assumption on the message size, this leaves a gap of κ in the upper and lower bounds. If we further assume that every message in authenticated protocols is signed, then the bounds match. It is an interesting open problem whether we can design an authenticated protocol that leaves most of the messages *unsigned* to do better than $O(\kappa n^2)$.

Byzantine Agreement. In Byzantine Agreement (BA), each party has an input value, and all parties try to decide on the same value. The requirement of BA is defined as follows.

▶ **Definition 6** (Byzantine Agreement (BA)). *A Byzantine agreement protocol must satisfy the following properties.*

1. *consistency: if two honest parties r and r' decide values v and v', then $v = v'$.*
2. *termination: every honest party decides a value and terminates.*
3. *validity: if all honest parties have the same input value, then all honest parties decide that value.*

Although our main focus of this paper is BA, we also mention a closely related problem called Byzantine broadcast (BB). In BB, a designated sender has an input to broadcast to all parties, and all parties try to decide on the same value. The requirement of BB is defined as follows.

▶ **Definition 7** (Byzantine Broadcast (BB)). *A Byzantine broadcast protocol must satisfy the following properties.*

1. *consistency: same as above.*
2. *termination: same as above.*
3. *validity: if the sender is honest, then all honest parties decide the sender's value.*

It is easy to transform a BA protocol into a BB protocol preserving the same resilience and quadratic communication complexity by having an initial round for the sender to broadcast its input value before starting the BA protocol [21]. As the Dolev-Reischuk lower bound holds for both BA and BB, our results establish the tightness of the quadratic communication complexity for BB as well (though the resilience $f < n/2$ is not optimal for BB, which is possible under any $f < n$).

3 Recursive Framework of Byzantine Agreement with Quadratic Communication

This section reviews the recursive framework to construct a BA protocol with quadratic communication introduced by Berman et al. [6] for $f < n/3$, and making it works for $f < n/2$.

Dissecting Berman et al. In the Berman et al. protocol, parties are partitioned into two halves, and each half runs the BA protocol recursively in sequential order. The partition continues until we reach a BA instance with a constant number of parties, where using any inefficient BA protocol will not impact the overall complexity. At each recursive step, additional quadratic communication is incurred besides the two recursive BA calls. It is not hard to see that the overall communication complexity is quadratic.

Since the fraction of faults in the entire parties is less than $1/3$, one of two halves also has faults of less than $1/3$ and thus achieve a "correct" BA. However, even if the first committee is correct, the potential incorrect second BA instance may "ruin" the result of the first one. To prevent this, parties run a few rounds of preprocessing steps called "universal exchange" in Berman et al. before each recursive BA call. The universal exchange step helps parties "stick to" a value (ignoring the recursive BA output) if all honest parties already agree on that value. In more detail, if the first run of recursive BA is correct and all honest parties agree on a value, the universal exchange before the second run makes sure all honest parties stick to it and the second run cannot change the agreed-upon value.

A tricky situation this universal exchange step needs to handle is when some honest parties stick to a value but other parties do not. In this case, this step needs to ensure that, if any honest party sticks to a value, other parties at least input that value to the subsequent BA call. The validity property of a correct recursive BA call will ensure agreement.

Here, the above recursive construction itself is independent of f, but the universal exchange step of Berman et al. relies on a quorum-intersection argument which only works under $f < n/3$. To make the framework independent of f, we abstract the functionality of this step as *graded Byzantine agreement* (GBA), since it is essentially the agreement version of graded broacast [16, 19]. In the rest of this section, we formally define the GBA primitive and construct a BA protocol using a GBA protocol as a black-box and prove its correctness.

3.1 Graded Byzantine Agreement

In graded Byzantine agreement (GBA), each party r has an input, and outputs a tuple (v, g) where v is the output value and $g \in \{0, 1\}$ is a grade bit.

▶ **Definition 8** (Graded Byzantine Agreement (GBA)). *A Graded Byzantine agreement protocol must satisfy the following properties.*

1. *consistency: if an honest party outputs* $(v, 1)$*, then all honest parties output* $(v, *)$*.*
2. *validity: if all honest parties have the same input value* v*, then all honest parties output* $(v, 1)$
3. *termination: every honest party outputs and terminates.*

The "stick to" nature is expressed by the grade bit g. The consistency property requires that if an honest party sticks to a value v, i.e., output v with $g = 1$, then all honest parties output the same value v. The validity property states that if all honest parties have the same input value v, they all stick to the value. These two properties capture what the universal exchange step needs to achieve explained at an intuitive level.

3.2 Recursive Construction of Byzantine Agreement

Next, we present the recursive BA protocol RBA in Figure 1. Let Q_w denote a set of parties that run a BA protocol. Since the protocol is recursive, the set Q_w is also defined recursively. Q_1 is a set of all n parties. Q_{2w} is the first $\lceil |Q_w|/2 \rceil$ parties in Q_w, and Q_{2w+1} is the remaining $\lfloor |Q_w|/2 \rfloor$ parties. All parties start by running $\mathsf{RBA}(Q_1)$ at the beginning.

RBA(Q_w)

Let r be a party. A variable v_r is initialized with the input value.

If $|Q_w| \leq M$, parties in Q_w execute any inefficient BA protocol with the $f < n/2$ resilience and output the decision of this inefficient BA. Otherwise, run the following steps, and output v_r and terminate.

1. Run GBA(Q_w) with input v_r. Let (v_r, g_r) be r's output from GBA(Q_w).
2. If $r \in Q_{2w}$, run RBA(Q_{2w}) with input v_r, and send the output v to all parties in Q_w. Otherwise, wait until the step finish.
3. If r receives the same value v from a majority of parties in Q_{2w} and $g_r = 0$, set v_r to v.
4. Run GBA(Q_w) with input v_r. Let (v_r, g_r) be r's output from GBA(Q_w).
5. If $r \in Q_{2w+1}$, run RBA(Q_{2w+1}) with input v_r, and send the output v to all parties in Q_w. Otherwise, wait until the step finish.
6. If r receives the same value v from a majority of parties in Q_{2w+1} and $g_r = 0$, set v_r to v.

Figure 1 Byzantine Agreement with $O(\kappa n^2)$ communication and $f < \frac{n}{2}$.

If the size of the RBA instance gets below a constant, denoted as M in the figure, parties can run any inefficient BA protocol with cubic or even higher communication complexity but with the desired resilience up to $f < n/2$. There are many such constructions in the literature [21, 14, 19, 2]; we do not describe these protocols. Otherwise, parties run two instances of RBA recursively to further reduce the instance size. Before each recursive call, they run a given GBA protocol denoted GBA. The grade bit output g_r of the GBA determines if a party r "sticks to" the GBA output or adopts the recursive RBA output.

Correctness of the Protocol. We prove the correctness of RBA for $f < n/2$ assuming the given GBA protocol GBA also tolerates $f < n/2$. The proof is easily extended for $f \leq (\frac{1}{2} - \varepsilon)n$. Below, minority faults within a set of parties Q mean at most $\lfloor (|Q| - 1)/2 \rfloor$ faults.

▶ **Lemma 9.** RBA *solves BA in the presence of minority faults.*

Proof. Termination is obvious. The proof for validity is also easy. If all honest parties have the same input value $v_r = v$, then due to the validity of GBA, all honest parties output $(v, 1)$ in step-1. Thus, they do not change v_r at step-3 and input $v_r = v$ into the GBA of step-4. Again due to the validity of GBA, all honest parties output $(v, 1)$ in step-4, do not change v_r at step-6, and all output v.

Next, we prove consistency. When $|Q| \leq M$, the correctness of RBA reduces to the correctness of the given inefficient BA. We just need to prove for the recursive step. Specifically, we will prove that RBA solves BA under n parties with minority faults, if RBA solves BA under $< n$ parties with minority faults.

Consider RBA(Q_w). Since Q_w has minority faults, at least one of the two halves Q_{2w} and Q_{2w+1} has minority faults. Let us first consider the case where Q_{2w} has minority faults. Here, there are two situations with regard to the result of step-1: (i) all honest parties in Q_w set g_r to 0, or (ii) at least an honest party in Q_w sets g_r to 1.

In the first situation, all honest parties will set v_r to the majority output of step-2. By the consistency and termination of RBA(Q_{2w}), all honest parties in Q_w receive the same value v from honest parties in Q_{2w} (which constitute a majority in Q_{2w}). Thus, all honest parties in Q_w set v_r to v in step-3.

In the second situation, since some honest party sets g_r to 1 in step-1, then by the consistency of GBA, all honest parties in Q_w set v_r to the same value v at the end of step-1. By the validity of RBA(Q_{2w}), all honest parties in Q_{2w} output v, so all honest parties in Q_w receive v from a majority of parties in Q_{2w}. Thus, all honest parties in Q_w set v_r to v in step-3.

Therefore, in both situations, all honest parties in Q_w have the same value $v_r = v$ at the beginning of step-4. Then, by the validity of GBA, all honest parties in Q_w set (v_r, g_r) to $(v, 1)$ in step-4, so will not change their v_r in step-6 and all output the same value v.

The other case where Q_{2w+1} has minority faults can be proved similarly. No matter which of the two situations holds at step-4 (all have $g_r = 0$ or some have $g_r = 1$), all honest parities in Q_w have the same value $v_r = v$ at the end of step-6 and output the same value v. Therefore, regardless of whether Q_{2w} or Q_{2w+1} has minority faults, consistency holds. ◀

With some foresight, we will construct GBA with quadratic communication in the later section. This will give RBA with quadratic communication in total.

▶ **Lemma 10** (Communication Complexity). *If the communication complexity of* GBA *is* $O(\kappa n^2)$*, then the communication complexity of* RBA *is* $O(\kappa n^2)$.

Proof. The communication complexity of RBA is given as a recurrence below. Let s be the number of parties in an RBA instance.

$$C(s) = \begin{cases} O(\kappa) & (\text{if } s \leq M) \\ C(\lfloor s/2 \rfloor) + C(\lceil s/2 \rceil) + O(\kappa s^2) & (\text{otherwise}) \end{cases}$$

For any n, the depth of the recursion k satisfies $2^{k-1}M \leq n \leq 2^k M$. Hence, $C(n) \leq 2^k O(\kappa) + \sum_{i=0}^{k} 2^i O(\kappa(n/2^i)^2) = O(\kappa n^2)$. ◀

4 Two Constructions of Graded Byzantine Agreement

This section presents two constructions of GBA protocols with different trade-offs to instantiate two BA protocols from the recursive framework in the previous section and complete the proof of Theorem 1 and 2.

4.1 Graded Byzantine Agreement with Threshold Signature Scheme

We first present a GBA protocol (denoted $\frac{1}{2}$-GBA) with quadratic communication and $f < n/2$ assuming a threshold signature scheme, which complete the proof of Theorem 1. We describe $\frac{1}{2}$-GBA in Figure 2. The parameter Q is a set of parties that participate in the protocol. Let $n = |Q|$.

Intuitive overviews. The construction is inspired by a few recent work on synchronous BB and BFT protocols [2, 3, 5]. Rounds 1–3 form a set of $n - f$ vote-1 (vote1-certificate) for the same value v, denoted $\mathcal{C}^1(v)$. Here, if an honest party votes for a value v in round 3, it must have received and multicast $n - f$ echo (echo-certificate) for v, denoted $\mathcal{E}(v)$ in round 2. Moreover, if a party receives a conflicting echo-certificate $\mathcal{E}(v')$ by the end of round 2, it does not vote in round 3. Therefore, rounds 1 and 2 prevent conflicting vote1-certificates from being created.

$\frac{1}{2}$-GBA(Q)

Let r be a party. $n = |Q|$, and $f < \lfloor (n-1)/2 \rfloor$. A variable v_r is initialized to the input value. g is initialized to 0. Run the following within the set of parties Q. $\langle x \rangle_r$ is a signature share on message x of a $(n-f, |Q|)$-threshold signature.

1. Multicasts $\langle \mathsf{echo}, v_r \rangle_r$.
2. If r receives $n-f$ $\langle \mathsf{echo}, v \rangle_*$, combine them into a threshold signature denoted $\mathcal{E}(v)$, and then multicasts $\mathcal{E}(v)$.
3. If r have multicast $\mathcal{E}(v)$ in round 2, and does not receive $\mathcal{E}(v')$ $(v' \neq v)$ by the end of round 2, multicasts $\langle \mathsf{vote\text{-}1}, v \rangle_r$.
4. If r receives $n-f$ $\langle \mathsf{vote\text{-}1}, v \rangle_*$, combine them into a threshold signature denoted $\mathcal{C}^1(v)$, and then multicasts $\mathcal{C}^1(v)$ and $\langle \mathsf{vote\text{-}2}, v \rangle_r$.
   At the end of the round, if r receives $\mathcal{C}^1(v)$, sets v_r to v. If r receives $n-f$ $\langle \mathsf{vote\text{-}2}, v \rangle_*$, denoted $\mathcal{C}^2(v)$, sets g to 1.

Finally, outputs (v_r, g).

Figure 2 Graded Byzantine agreement with $f < n/2$ with a threshold signature scheme.

Round 4 forms a set of $n-f$ vote-2 (vote2-certificate) for a value v, denoted $\mathcal{C}^2(v)$. If a party receives a vote1-certificate $\mathcal{C}^1(v)$ by the end of round 3, it sends vote-2 for a value v (along with $\mathcal{C}^1(v)$) in round 4. Therefore, if a vote2-certificate $\mathcal{C}^2(v)$ is formed, all honest parties can receive a vote1-certificate $\mathcal{C}^1(v)$.

Finally, a party outputs a value v if it receives a vote1-certificate $\mathcal{C}^1(v)$, and it further sets the grade bit g to 1 if it also receives a vote2-certificate $\mathcal{C}^2(v)$. Consistency follows from the properties above. Moreover, if all honest parties have the same input value v, all honest parties (at least $n-f$) receive both $\mathcal{C}^1(v)$ and $\mathcal{C}^2(v)$ and output $(v, 1)$, so validity also holds.

Correctness of the protocol. We prove the correctness of $\frac{1}{2}$-GBA assuming $f < n/2$. The termination of $\frac{1}{2}$-GBA is trivial, and thus we prove the consistency and validity.

▶ **Lemma 11.** *If $\mathcal{C}^1(v)$ and $\mathcal{C}^1(v')$ are both created, then $v = v'$.*

Proof. Suppose $\mathcal{C}^1(v)$ is created, then at least an honest party r must have multicast vote-2 for v in round 3. That implies r received $\mathcal{E}(v)$ and multicast it in round 2. Then, all honest parties must have received $\mathcal{E}(v)$ by round 3, and all honest parties could not have multicast vote-2 for $v' \neq v$. Therefore, $\mathcal{C}^1(v')$ cannot be created unless $v' = v$. ◀

▶ **Lemma 12** (Consistency). *If an honest party outputs $(v, 1)$, then all honest parties output $(v, *)$*

Proof. Suppose an honest party outputs $(v, 1)$, then it must have received $\mathcal{C}^2(v)$ for a value v by the end of round 4. Then, at least one honest party must have multicast $\mathcal{C}^1(v)$ in round 4, and all honest parties must have received it by the end of round 4. Since there is not $\mathcal{C}^1(v')$ for a different value v' by Lemma 11, all honest parties set v_r to v at the end of round 4 and thus output v. ◀

▶ **Lemma 13** (Validity). *If all honest parties have the same input value v, then all honest parties output $(v, 1)$.*

Proof. If all honest parties have the same input value v, they all multicast $\langle \mathsf{echo}, v\rangle$ in round 1, and thus $\mathcal{E}(v)$ should be formed and $\mathcal{E}(v')$ for $v' \neq v$ cannot be formed. In the same way, all honest parties multicast $\langle \mathsf{vote\text{-}1}, v\rangle$ in round 3 and $\langle \mathsf{vote\text{-}2}, v\rangle$ in round 4. Therefore, $\mathcal{C}^1(v)$ and $\mathcal{C}^2(v)$ should be formed and $\mathcal{C}^1(v')$ and $\mathcal{C}^2(v')$ for $v' \neq v$ cannot be formed. Thus, all honest parties output $(v, 1)$. ◀

Communication complexity and discussion. With threshold signatures, all certificates $\mathcal{E}(v)$, $\mathcal{C}^1(v)$, $\mathcal{C}^2(v)$ are $O(\kappa)$ in size, and the commutation complexity of $\frac{1}{2}$-GBA is clearly $O(\kappa n^2)$. But we note that the RBA protocol in Figure 1 invokes GBA with different numbers of participants for each depth in the recursion and hence requires different thresholds for threshold signatures. As a result, each party needs $\Theta(\log n)$ key setups.

4.2 Graded Byzantine Agreement without Threshold Signature Scheme

Next, we present a GBA protocol (denoted $(\frac{1}{2} - \varepsilon)$-GBA) with quadratic communication and $f \leq (\frac{1}{2} - \varepsilon)n$ for any positive constant ε without relying on any threshold signature scheme or trusted setup (beyond the standard PKI). We describe $(\frac{1}{2} - \varepsilon)$-GBA in Figure 3.

Intuitive overview. The main motivation of $(\frac{1}{2} - \varepsilon)$-GBA is to remove the use of threshold signature. Thus, let us first review why threshold signature scheme is necessary in the GBA protocol from the previous section. The threshold signature scheme is used to aggregate a set of $n - f$ signatures (quorum certificate $\mathcal{E}(v)$ in round 2 and vote1-certificate $\mathcal{C}^1(v)$ in round 4). If these are not aggregated, each party needs to multicast linear-sized certificates, leading to cubic communication in total.

Therefore, to remove aggregation while keeping the communication quadratic, we need to remove multicast. However, multicasting quorum certificates in round 2 and 4 is key to consistency. Specifically, multicasting an echo-certificate $\mathcal{E}(v)$ in round 2 helps honest parties detect a conflicting echo-certificate $\mathcal{E}(v')$, which allows honest parties to decide the value v safely; multicasting a vote1-certificate $\mathcal{C}^1(v)$ in round 4 helps notify all honest parties of the existence of $\mathcal{C}^1(v)$, which allows the party to decide the value v with confidence, i.e., grade bit $g = 1$.

Our key new technique is to replace the multicast steps with more efficient yet robust dissemination of certificates through a predetermined expander graph with a constant degree.

▶ **Definition 14** (Expander). *An (n, α, β)-expander $(0 < \alpha < \beta < 1)$ is a graph of n vertices such that, for any set S of αn vertices, the number of neighbors of S is more than βn.*

It is well-known that for any n and $0 < \alpha < \beta < 1$, (n, α, β)-expanders exist. For our purpose, we need an $(n, 2\varepsilon, 1 - 2\varepsilon)$-expander; in other words, we set $\alpha = 2\varepsilon$ and $\beta = 1 - 2\varepsilon$. Henceforth, we write an $(n, 2\varepsilon, 1 - 2\varepsilon)$-expander as $G_{n,\varepsilon}$. For completeness, we show in Appendix A that for all positive ε and for all n, the required expander $G_{n,\varepsilon}$ always exists.

Instead of sending a quorum certificate to all other parties, a party propagates it to a constant number of neighbors in $G_{n,\varepsilon}$. Therefore, the total number of messages is reduced from quadratic to linear, and thus the total communication is kept quadratic even though some messages contain a linear number of signatures. Our key observation is that although the message is not sent to everyone (since the expander is not a fully connected graph), it is sufficient to maintain consistent decisions among honest parties.

In more detail, in round 3, each party multicasts vote-1 for a value v only if it propagated an echo-certificate $\mathcal{E}(v)$ in round 2 and it does not receive a conflicting echo-certificate $\mathcal{E}(v')$. If a vote1-certificate $\mathcal{C}^1(v)$ forms, at least $n - 2f = 2\varepsilon n$ are honest. They must have

$(\frac{1}{2}-\varepsilon)$-GBA$(Q)$

Let r be a party. $n = |Q|$, and $f = \lfloor(\frac{1}{2}-\varepsilon)n\rfloor$. A variable v_r is initialized to the input value. g is initialized to 0. "Propagate" means sending to all neighbors in $G_{n,\varepsilon}$ and "multicast" means sending to all n parties. Run the following within the set of parties Q. $\langle x\rangle_r$ is a digital signature on a message x.

1. Multicasts $\langle \mathsf{echo}, v_r\rangle_r$.
2. If r receives $n-f$ $\langle \mathsf{echo}, v\rangle_*$, denoted $\mathcal{E}(v)$, propagates $\mathcal{E}(v)$.
3. If r have propagated $\mathcal{E}(v)$ in round 2, and does not receive $\mathcal{E}(v')$ $(v' \neq v)$ by the end of round 2, multicasts $\langle \mathsf{vote\text{-}1}, v\rangle_r$.
4. If r receives $n-f$ $\langle \mathsf{vote\text{-}1}, v\rangle_*$, denoted $\mathcal{C}^1(v)$, propagate $\mathcal{C}^1(v)$, and multicasts $\langle \mathsf{vote\text{-}2}, v\rangle_r$. .
5. If r receives $\mathcal{C}^1(v)$ by the end of round 4, multicasts $\langle \mathsf{vote\text{-}3}, v\rangle_r$.
   At the end of the round, if r receives $f+1$ $\langle \mathsf{vote\text{-}3}, v\rangle_*$, sets v_r to v. If r receives $n-f$ $\langle \mathsf{vote\text{-}2}, v\rangle_*$, denoted $\mathcal{C}^2(v)$, set g to 1.

Finally, outputs (v_r, g).

Figure 3 Graded Byzantine agreement with $f \leq (\frac{1}{2}-\varepsilon)n$ without threshold signature scheme.

propagated $\mathcal{E}(v)$ and it will be received by more than $(1-2\varepsilon)n = 2f$ parties. Out of these, at least $f+1$ are honest and will not vote for a conflicting value. This guarantee the unique existence of vote1-certificate $\mathcal{C}^1(v)$.

Confirming the existence of a vote1-certificate is trickier as we cannot afford multicasts to notify all parties. We achieve this in two steps. In round 4, after propagating $\mathcal{C}^1(v)$, the party multicast vote-2 for v. If a vote2-certificate $\mathcal{C}^2(v)$ forms, due to the expansion property, at least $f+1$ honest parties receive $\mathcal{C}^1(v)$ by the end of round 4. Then, in round 5, if a party receives $\mathcal{C}^1(v)$, it multicast vote-3 message for v. As at least $f+1$ honest parties receives $\mathcal{C}^1(v)$, all honest parties can receive $f+1$ vote-3 message for v, which works as a succinct proof of existence of $\mathcal{C}^1(v)$. This allows all honest parties to confirm the existence of a vote1-certificate.

Correctness of the protocol. We prove the correctness of $(\frac{1}{2}-\varepsilon)$-GBA assuming $f \leq (\frac{1}{2}-\varepsilon)n$ for any positive constant ε. The termination of $(\frac{1}{2}-\varepsilon)$-GBA is trivial, and thus we prove the consistency and validity.

▶ **Lemma 15.** *If $\mathcal{C}^1(v)$ and $\mathcal{C}^1(v')$ are both created, then $v = v'$.*

Proof. Suppose $\mathcal{C}^1(v)$ is created, then at least $2\varepsilon n$ honest parties must have propagated $\mathcal{E}(v)$ in round 2. Then, due to the expansion property of $G_{n,\varepsilon}$, more than $2f$ parties, out of which at least $f+1$ honest parties must have received $\mathcal{E}(v)$ by the end of round 2, and do not send $\langle \mathsf{vote\text{-}1}, v'\rangle_*$ for a different value $v' \neq v$ in round 3. Therefore, $\mathcal{C}^1(v')$ cannot be created unless $v' = v$. ◀

▶ **Lemma 16** (Consistency). *If an honest party outputs $(v, 1)$, then all honest parties output $(v, *)$.*

Proof. Suppose an honest party outputs $(v, 1)$, then it must have received $\mathcal{C}^2(v)$ for a value v by the end of round 5. Then, at least $2\varepsilon n$ honest parties must have propagated $\mathcal{C}^1(v)$ in round 4. Due to the expansion property of $G_{n,\varepsilon}$, more than $2f$ parties, out of which at least $f+1$ honest parties must have received $\mathcal{C}^1(v)$ by the end of round 4, and multicast $\langle \mathsf{vote\text{-}3}, v\rangle_*$ in round 5. Thus, all honest parties must have received $f+1$ $\langle \mathsf{vote\text{-}3}, v\rangle_*$ by the end of round 5. Here, as $\mathcal{C}^1(v')$ for a different value $v' \neq v$ cannot form by Lemma 15, honest parties could not have multicast $\langle \mathsf{vote\text{-}3}, v'\rangle_*$. Therefore, all honest party could not have received $f+1$ $\langle \mathsf{vote\text{-}3}, v'\rangle_*$, and thus output v. ◀

▶ **Lemma 17** (Validity). *If all honest parties have the same input value v, then all honest parties output $(v, 1)$*

Proof. If all honest parties have the same input value v, they all multicast $\langle \mathsf{echo}, v\rangle$ in round 1, and thus $\mathcal{E}(v)$ must form and $\mathcal{E}(v')$ for $v' \neq v$ cannot form. Then, all honest parties multicast $\langle \mathsf{vote\text{-}1}, v\rangle$ in round 3, propagate $\mathcal{C}^1(v)$ and multicast $\langle \mathsf{vote\text{-}2}, v\rangle$ in round 4, and $\langle \mathsf{vote\text{-}3}, v\rangle$ in round 5. Therefore, all honest parties receive both $\mathcal{C}^2(v)$ and $f+1$ $\langle \mathsf{vote\text{-}3}, v\rangle_*$, and output $(v, 1)$. ◀

Communication complexity. All certificates $\mathcal{E}(v)$, $\mathcal{C}^1(v)$, $\mathcal{C}^2(v)$ are $O(\kappa n)$ in size, but are only sent through the degree-d expander. All the multicasted messages are $O(\kappa)$ in size. Thus, the communication complexity of $(\frac{1}{2}-\varepsilon)$-GBA is $O(\kappa n^2 d)$. Appendix A shows that $d = O(\frac{1}{\varepsilon})$ suffices, so the communication complexity of $(\frac{1}{2}-\varepsilon)$-GBA is $O(\kappa n^2)$ when ε is a constant, and is $O(\kappa n^2/\varepsilon)$ in general. This communication complexity is inherited in the RBA protocol in Figure 1.

5 Conclusion

In this paper, we provided two results: (1) a BA protocol with quadratic communication with optimal resilience $f < n/2$ with a trusted setup, and (2) a BA protocol with quadratic communication with near optimal resilience $f \leq (\frac{1}{2}-\varepsilon)n$ without trusted setup. Even with our new results, the tightness of the Dolev-Reischuk lower bound is still open for some settings, for exmaple, BA under a standard PKI model with $(\frac{1}{2}-\varepsilon)n < f < n/2$, or quadratic BB with $f \geq n/2$ even with a trusted setup. These are intriguing open questions for future work.

References

1 Ittai Abraham, TH Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 317–326, 2019.

2 Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. Synchronous byzantine agreement with expected $o(1)$ rounds, expected $o(n^2)$ communication, and optimal resilience. In *Financial Cryptography and Data Security (FC)*, pages 320–334. Springer, 2019.

3 Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Maofan Yin. Sync hotstuff: Simple and practical synchronous state machine replication. In *IEEE Symposium on Security and Privacy (S&P)*, pages 106–118. IEEE, 2020.

4 Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. Validated asynchronous byzantine agreement with optimal resilience and asymptotically optimal time and word communication. *arXiv preprint*, 2018. `arXiv:1811.01332`.

5 Ittai Abraham, Kartik Nayak, Ling Ren, and Zhuolun Xiang. Optimal good-case latency for byzantine broadcast and state machine replication. *arXiv preprint*, 2020. `arXiv:2003.13155`.

6 Piotr Berman, Juan A Garay, and Kenneth J Perry. Bit optimal distributed consensus. In *Computer science*, pages 313–321. Springer, 1992.

7 Christian Cachin, Klaus Kursawe, Frank Petzold, and Victor Shoup. Secure and efficient asynchronous broadcast protocols. In *Annual International Cryptology Conference (CRYPTO)*, pages 524–541. Springer, 2001.

8 Christian Cachin and Stefano Tessaro. Asynchronous verifiable information dispersal. In *IEEE Symposium on Reliable Distributed Systems (SRDS)*, pages 191–201. IEEE, 2005.

9 Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *3rd Symposium on Operating Systems Design and Implementation (OSDI)*, pages 173–186. USENIX, 1999.

10 Jing Chen and Silvio Micali. Algorand. *arXiv preprint*, 2016. `arXiv:1607.01341`.

11 Shir Cohen, Idit Keidar, and Alexander Spiegelman. Not a coincidence: Sub-quadratic asynchronous byzantine agreement whp. *arXiv preprint*, 2020. `arXiv:2002.06545`.

12 Danny Dolev, Michael J Fischer, Rob Fowler, Nancy A Lynch, and H Raymond Strong. An efficient algorithm for byzantine agreement without authentication. *Information and Control*, 52(3):257–274, 1982.

13 Danny Dolev and Rüdiger Reischuk. Bounds on information exchange for byzantine agreement. *Journal of the ACM (JACM)*, 32(1):191–204, 1985.

14 Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

15 Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM*, 35(2):288–323, 1988.

16 Paul Feldman and Silvio Micali. Optimal algorithms for byzantine agreement. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 148–161, 1988.

17 Michael J Fischer, Nancy A Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Computing*, 1(1):26–39, 1986.

18 Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

19 Jonathan Katz and Chiu-Yuen Koo. On expected constant-round protocols for byzantine agreement. *Journal of Computer and System Sciences*, 75(2):91–112, 2009.

20 Valerie King and Jared Saia. Breaking the $o(n^2)$ bit barrier: scalable byzantine agreement with an adaptive adversary. *Journal of the ACM (JACM)*, 58(4):1–24, 2011.

21 Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

22 Yuan Lu, Zhenliang Lu, Qiang Tang, and Guiling Wang. Dumbo-mvba: Optimal multi-valued validated asynchronous byzantine agreement, revisited. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 129–138, 2020.

23 Silvio Micali. Byzantine agreement, made trivial, 2016.

24 Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In *ACM Conference on Computer and Communications Security (CCS)*, pages 31–42, 2016.

25 Kartik Nayak, Ling Ren, Elaine Shi, Nitin H Vaidya, and Zhuolun Xiang. Improved extension protocols for byzantine broadcast and agreement. *arXiv preprint*, 2020. `arXiv:2002.11321`.

26 Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.

27 Alexander Spiegelman. In search for a linear byzantine agreement. *arXiv preprint*, 2020. `arXiv:2002.06993`.

28 Georgios Tsimos, Julian Loss, and Charalampos Papamanthou. Nearly quadratic broadcast without trusted setup under dishonest majority. *IACR Cryptology ePrint Archive, Report 2020/894*, 2020.

29 Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus with linearity and responsiveness. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 347–356, 2019.

A Expander

We show that an expander G_ε in the Definition 14 exists for all positive constant ε. We use $\Gamma(V, G)$ to denote a set of all neighbors of V in a graph G.

▶ **Theorem 18** (Existence of Expander). *For all positive integer n and positive ε, there exists an expander $G_{n,\varepsilon}$ with degree $d = O(\frac{1}{\varepsilon})$.*

Proof. Let $c = 2\varepsilon$ and $G_{n,\varepsilon}$ is an $(n, c, 1-c)$-expander. For $c \geq 1/2$, the expansion property becomes trivial. Note that for our purpose, we can consider a vertex a neighbor of itself, so a graph with a self edge for every vertex is an $(n, c, 1-c)$-expander for $c \geq 1/2$. So we just need to focus on $c < 1/2$.

Consider a random d degree graph G taking the union of random d perfect matchings (if n is odd, the first party has two links). In each perfect matching P, for any set of cn parties (say S), and any set of $(1-c)n$ parties (say T), the probability that $\Gamma(S, P) \subseteq T$ is at most

$$\Pr[\Gamma(S,P) \subseteq T] \leq \left(\frac{(1-c)n}{n}\right)^{\frac{cn}{2}} = (1-c)^{\frac{cn}{2}}.$$

Thus, the probability that any set of cn parties do not expand in the graph, i.e., $|\Gamma(S, G)| \leq (1-c)n$ for any S, is at most

$$\begin{aligned} & \binom{n}{cn}\binom{n}{(1-c)n}(1-c)^{\frac{cdn}{2}} \\ \leq\ & \left(\frac{e}{c}\right)^{cn}\left(\frac{e}{1-c}\right)^{(1-c)n}(1-c)^{\frac{cdn}{2}} \\ \leq\ & \left(e\left(\frac{1}{c}\right)^{c}\left(\frac{1}{1-c}\right)^{1-c}(1-c)^{\frac{cd}{2}}\right)^{n} \end{aligned}$$

The above probability upper bound is smaller than 1 (in fact, exponentially small in n), when the degree d is sufficiently large. The precise requirement on d is $\frac{d}{2} > \frac{1}{c} - 1 + \frac{c\log c - \log e}{c\log(1-c)}$. It is not hard to show that when $0 < c < \frac{1}{2}$, $d = O(\frac{1}{c}) = O(\frac{1}{\varepsilon})$ will suffice. This means there is a non-zero (in fact, overwhelmingly large) probability that a randomly chosen graph is an expander. Thus, $G_{n,\varepsilon}$ with degree $d = O(\frac{1}{\varepsilon})$ exists. ◀