Round Complexity of Authenticated Broadcast with a Dishonest Majority*

Juan A. Garay[†] Jonathan Katz[‡] Chiu-Yuen Koo[‡] Rafail Ostrovsky[§]

Abstract

Broadcast among n parties in the presence of $t \geq n/3$ malicious parties is possible only with some additional setup. The most common setup considered is the existence of a PKI and secure digital signatures, where so-called authenticated *broadcast is achievable for any $t < n$.*

It is known that $t + 1$ rounds are necessary and sufficient for deterministic *protocols achieving authenticated broadcast. Recently, however,* randomized *protocols running in expected* constant *rounds have been shown for the case of $t < n/2$. It has remained open whether randomization can improve the round complexity when an honest majority is not present. We address this question and show upper/lower bounds on how much randomization can help:*

- *For $t \leq n/2 + k$, we show a randomized broadcast protocol that runs in expected $\mathcal{O}(k^2)$ rounds. In particular, we obtain expected constant-round protocols for $t = n/2 + \mathcal{O}(1)$.*
- *On the negative side, we show that even randomized protocols require $\Omega(2n/(n-t))$ rounds. This in particular rules out expected constant-round protocols when the fraction of honest parties is sub-constant.*

1. Introduction

Designing protocols for simulating a broadcast channel over a point-to-point network in the presence of faults is a fundamental problem in distributed computing and cryptography. Much work has focused both on characterizing the *feasibility* of protocols for solving the problem in different settings, as well as on the inherent *round complexity* of such protocols. In a synchronous network with pairwise authenticated channels and no additional setup, the classical results of Pease, Shostak, and Lamport [24, 29] show that broadcast among n parties is achievable if and only if the number of malicious parties t satisfies $t < n/3$. In this setting, a lower bound of $t + 1$ rounds for any deterministic protocol is known [16]. A protocol with this round complexity — but with exponential message complexity — was shown in the initial work by Pease et al. [24, 29]. Following a long sequence of works [9, 1, 33, 12, 26, 5, 4], Garay and Moses [19] showed a deterministic, polynomial-time Byzantine agreement protocol having optimal resilience $t < n/3$ and optimal round complexity $t + 1$.

To circumvent the above-mentioned lower bound on the round complexity (as well as impossibility results for asynchronous networks [15]), researchers beginning with Rabin [31] and Ben-Or [2] explored the use of *randomization*. (See [7] for an early survey on the subject.) This culminated in the work of Feldman and Micali [14], who showed a broadcast protocol with optimal resilience that runs in expected constant rounds.[1]

To achieve resilience $t \geq n/3$, additional assumptions are needed even if randomization is used. The most common assumptions are the existence of digital signatures and the presence of a public-key infrastructure (PKI) established among the n parties in the network; this is referred to as the *authenticated* setting. Pease et al. [29, 24] showed an authenticated broadcast protocol for any $t < n$, and a polynomial-time protocol achieving this resilience was given by Dolev and Strong [13].

The $(t + 1)$-round lower bound for deterministic protocols holds in the authenticated setting as well [13], and the known protocols [29, 24, 13] meet this bound. Randomized protocols running in expected constant rounds for $t < n/2$ have been shown by Fitzi and Garay [17] (based on [6, 28]) under specific number-theoretic assumptions, and by Katz and Koo [23] based on signatures and a PKI alone.

When an honest majority is *not* available (i.e., $t \geq n/2$),

*A portion of this work was done while the authors were visiting the Institute for Pure and Applied Mathematics (IPAM), UCLA.

[†]Bell Labs, Alcatel-Lucent. E-mail: `garay@research.bell-labs.com`.

[‡]Dept. of Computer Science, University of Maryland. E-mail: `{jkatz,cykoo}@cs.umd.edu`. Work of J.K. supported in part by NSF CAREER award #0447075 and US-Israel Binational Science Foundation grant #2004240.

[§]Depts. of Computer Science and Mathematics, UCLA. E-mail: `rafail@cs.ucla.edu`. Work supported in part by an IBM Faculty Award, a Xerox Innovation Group Award, NSF Cybertrust grant #0430254, and a U.C. MICRO grant.

[1]The Feldman-Micali protocol requires private channels. Goldwasser et al. [21] show a broadcast protocol for $t \leq n/(3+\epsilon)$ that runs in expected $\mathcal{O}(\log n)$ rounds and does not require private channels.

there has been no progress since the initial work of [29, 24, 13] on improving the round complexity of authenticated broadcast.[2] Besides being an interesting and fundamental problem in its own right, authenticated broadcast is often used as a sub-routine within larger protocols that are designed and analyzed using the abstraction that a broadcast channel exists. For the specific case of secure multi-party computation with a dishonest majority, we remark that although meaningful security notions can be achieved even without broadcast [20], and fairness cannot be achieved even with broadcast [8], there are still advantages to having broadcast available. Specifically, broadcast can be used to achieve *unanimous abort* [20], or *partial* notions of fairness [18, 22]. In contrast, the constant-round "broadcast-with-abort" protocol of [20] does not appear to suffice for such applications.

Our contributions. In this paper we make the first progress toward characterizing when randomized protocols can beat the $(t+1)$-round barrier for $t \geq n/2$.

- We show a randomized broadcast protocol tolerating $t \leq n/2 + k$ malicious parties that terminates in an expected $\mathcal{O}(k^2)$ rounds. This is an improvement over existing state of the art for $t = n/2 + o(\sqrt{n})$, and gives an expected constant-round protocol when $t = n/2 + \mathcal{O}(1)$.
- We show that no randomized broadcast protocol tolerating t malicious parties terminates in $2n/(n-t)-2$ or fewer rounds. This in particular means that when the fraction of honest parties is sub-constant, it is impossible to obtain protocols with expected constant round complexity. It also implies that the Dolev-Strong protocol [13] has optimal round complexity (to within a constant factor) when $t = n - \mathcal{O}(1)$.

Organization. In Section 2.1, we describe our model and give the standard definitions of broadcast and Byzantine agreement. We present the technical tools we use in Section 2.2; these include a generalization of gradecast [14] that may be of independent interest. We present our new broadcast protocol in Section 3, and prove our impossibility result in Section 4. Some proofs are deferred to the Appendix.

2. Preliminaries

2.1. Model and Definitions

We assume a standard point-to-point network in which parties $P_1, P_2, \ldots, P_n$ communicate in synchronous rounds using pairwise private and authenticated channels. When we say a protocol tolerates t dishonest parties, we always mean that it is secure against a *rushing* adversary who may *adaptively* corrupt up to t parties during execution of the protocol and coordinate the actions of these parties as they deviate from the protocol in an arbitrary manner.[3] Parties not corrupted by the adversary are called *honest*.

The existence of a PKI means that prior to execution of the protocol all parties hold the same vector $(pk_1, \ldots, pk_n)$ of public keys for a digital signature scheme, and each honest party P_i holds the honestly generated secret key sk_i associated with pk_i. When we describe signature computation in our protocols, we omit for simplicity certain additional information that should be signed along with the message. That is, when we say that party P_i signs message m and sends it to P_j, we implicitly mean that P_i signs the concatenation of m with additional information such as: (1) the identity of the recipient P_j, (2) the current round number, (3) an identifier for the message (in case multiple messages are sent to P_j in the same round); and (4) an identifier for the particular (sub-)protocol to which m belongs (in case multiple sub-protocols are being run; cf. [25]). This information is also verified, as appropriate, when the signature is verified.

We assume in our proofs that the adversary cannot forge valid signatures on behalf of honest parties. Using a standard hybrid argument and assuming the existence of one-way functions [27, 32], this implies that our protocols are secure against any computationally-bounded adversary. (Alternately, if stronger setup is assumed then information-theoretic pseudo-signatures [30] can be used.)

We now give the standard definition of broadcast [24].

Definition 1 (Broadcast). *A protocol for parties $\mathcal{P} = \{P_1, \ldots, P_n\}$, where a distinguished sender $P^* \in \mathcal{P}$ holds an initial input m, is a* broadcast protocol tolerating t malicious parties *if the following conditions hold for any adversary controlling at most t parties:*

Agreement: *All honest parties output the same value.*

Validity: *If the sender is honest, then all honest parties output m.* $\diamondsuit$

We will also rely on protocols for the related task of *Byzantine agreement* (BA). Here, each party holds an initial input: the agreement condition remains the same as above; validity requires that if all honest parties hold initial input m, then all honest parties will output m. Note that BA is impossible to achieve for $t \geq n/2$ (in any setting).

[2] The techniques used for $t < n/2$ do not immediately translate to the case of $t \geq n/2$: a key building block in the former setting is *verifiable secret sharing*, which is not even feasible in the latter setting.

[3] A *rushing* adversary waits until it receives messages from all honest parties in a given round before sending any messages of its own for that round. *Adaptive* corruption means that the adversary is allowed to corrupt parties on the fly, as opposed to deciding which parties to corrupt before execution of the protocol begins.

2.2. Tools

We describe two technical tools we use to construct our randomized broadcast protocol.

BA in expected constant rounds for $t < n/2$. The work of Katz and Koo [23] gives an authenticated BA protocol $\texttt{BA}_{\text{HonestMaj}}$ tolerating any $t < n/2$ malicious parties and running in expected constant rounds. Protocol $\texttt{BA}_{\text{HonestMaj}}$ satisfies the following stronger property that we will rely on in the present work:

Lemma 1. *If $h > n/2$ honest parties start $\texttt{BA}_{\text{HonestMaj}}$ with the same input, then all honest parties terminate protocol $\texttt{BA}_{\text{HonestMaj}}$ in* **exactly** *K rounds for some constant K.*

Gradecast. *Gradecast*, a generalization of *crusader agreement* [11], was introduced by Feldman and Micali [14]. As opposed to broadcast, where the honest parties are required to reach a unanimous decision, in gradecast the honest parties are allowed to disagree by "a small amount". Specifically, parties now output a *grade* along with their output value; the grade output by a party can be viewed as the "confidence" of this party in the sender. The gradecast protocol given by Feldman and Micali supports the three grades $\{0, 1, 2\}$, and runs in three rounds. Here, we generalize their protocol to the case of an arbitrary number of grades. We first present the definition:

Definition 2 (Gradecast with multiple grades)**.** *A protocol for parties $\mathcal{P} = \{P_1, \ldots, P_n\}$, where $P^* \in \mathcal{P}$ holds an initial input m, is a g^*-*gradecast protocol *(tolerating $n-1$ malicious parties) if the following conditions hold for any adversary controlling any number of parties:*

Functionality: *An honest party P_i outputs a message m_i and a grade $g_i \in \{0, 1, \ldots, g^*\}$.*

Correctness: *If the sender is honest, then $m_i = m$ and $g_i = g^*$ for all honest parties P_i.*

Soundness: *Let P_i, P_j be any two honest parties. If $g_i \geq 2$, then $m_j = m_i$ and $g_j \geq g_i - 1$. If $g_i = 1$, then $m_j = m_i$ or $g_j = 0$.* $\diamondsuit$

A similar primitive called "proxcast" was defined and constructed by Considine et al. [10]. Our construction differs from theirs in two ways. First, our construction is in the authenticated setting while theirs relies on the existence of "k-cast channels". Second, our protocol can tolerate any number of dishonest parties, while theirs only tolerates a constant fraction (the exact constant depends on the value of k) of malicious participants.

We now demonstrate a construction of g^*-gradecast for any value g^*. Specifically, we define a protocol $\texttt{M-Gradecast}(m, g^*)$ where m represents the initial value of the sender and g^* denotes the maximum supported grade. In the description that follows, each party P_i starts with internal variables $\bar{g}_i$, S_i, and m_i initialized to 0, the empty set, and $\perp$, respectively.

Protocol $\texttt{M-Gradecast}(m, g^*)$

Round 1: The sender computes a signature σ on m and sends (m, σ) to all parties.

Round 2 to **Round** $2g^* + 1$:

Step (a) Each party P_i does as follows: For each tuple (m', σ') received by the end of the previous round, if σ' is a valid signature by the sender on m' and $m' \notin S_i$, then:

- Set $S_i := S_i \cup \{m'\}$. If $|S_i| = 1$, then set $m_i := m'$.
- P_i sends (m', σ') to all other parties.

Step (b) If $(m_i \neq \perp)$ and $(|S_i| = 1)$ then set $\bar{g}_i := \bar{g}_i + 1$.

Output determination: Each party P_i sets $g_i := \lfloor \bar{g}_i/2 \rfloor$ and outputs (m_i, g_i).

Lemma 2. *Protocol $\texttt{M-Gradecast}(\cdot, g^*)$ is a g^*-gradecast protocol with round complexity $2g^* + 1$.*

The proof is given in the Appendix.

3. Randomized Broadcast Protocols for Dishonest Majority

As a warm-up, we first construct an expected constant-round broadcast protocol for the special case of $t = n/2$ (and n even) before dealing with the more general case.

3.1. The Case $t = n/2$

The main idea here is as follows: in the first phase, the sender will gradecast its input m. If the sender is honest, this gradecast is already enough to implement broadcast; on the other hand, if the other parties catch the sender cheating then they can exclude the sender and determine their output by executing $\texttt{BA}_{\text{HonestMaj}}$. The key point is that in the latter case, assuming $t = n/2$ to begin with, an *honest majority* is present once the dishonest dealer is excluded. (Variants of this idea — i.e., executing a protocol until either something good happens or some dishonest parties can be excluded — have been used in prior work on Byzantine agreement [1, 26, 5, 19].) Of course, we need to handle the scenario where some parties believe the sender is honest while other parties catch the sender cheating; this can be done using the grades obtained in the initial gradecast. We now provide a formal description of the protocol:

Phase I P^*, who holds input m, acts as the sender in an execution of `M-Gradecast`$(m, 2)$, outputs m, and then exits the protocol. Let (m_i, g_i) denote the output of P_i in this step.

Phase II All parties except P^* (who has already exited the protocol) run $\mathtt{BA}_{\text{HonestMaj}}$ in the following way:

- If $g_i = 2$, then P_i enters protocol $\mathtt{BA}_{\text{HonestMaj}}$ with input m_i, terminates $\mathtt{BA}_{\text{HonestMaj}}$ after K rounds (where K is the constant from Lemma 1), and outputs m_i. We stress that P_i outputs m_i regardless of the output (if any) of protocol $\mathtt{BA}_{\text{HonestMaj}}$.
- Otherwise (i.e., $g_i < 2$), P_i enters protocol $\mathtt{BA}_{\text{HonestMaj}}$ with input m_i, runs $\mathtt{BA}_{\text{HonestMaj}}$ until successful termination of the protocol, and outputs whatever directed to by $\mathtt{BA}_{\text{HonestMaj}}$.

We now argue that the above protocol achieves broadcast for $t = n/2$ in expected constant rounds. If the sender is honest then, by the correctness property of `M-Gradecast`$(m, 2)$, each honest party P_i outputs $(m_i = m, g_i = 2)$ in Phase I and thus, in Phase II, outputs $m_i = m$ after executing $\mathtt{BA}_{\text{HonestMaj}}$ for exactly K rounds. As the round complexity of Phase I is constant, the entire protocol runs for a strict constant number of rounds.

If the sender is dishonest, then protocol $\mathtt{BA}_{\text{HonestMaj}}$ is run with an honest majority. There are two sub-cases to consider. The first sub-case is that there exists an honest party P_i whose output in Phase I is $(m_i, g_i = 2)$. Then by the soundness property of `M-Gradecast`$(m, 2)$, all honest parties P_j have $m_j = m_i$. Hence all honest parties enter protocol $\mathtt{BA}_{\text{HonestMaj}}$ holding the same input m_i, and the protocol $\mathtt{BA}_{\text{HonestMaj}}$ terminates after K rounds with each honest party P_j outputting m_i, regardless of the grade g_j it output in the first step. The second sub-case is when all honest parties output a grade less than 2 in Phase I. Then all honest parties run $\mathtt{BA}_{\text{HonestMaj}}$ until termination, and so all honest parties output the same value in expected constant rounds.

3.2. The Case $t \leq n/2 + k$

In this section we construct a broadcast protocol `Rand-Bcast` for $t \leq n/2 + k$ that runs in expected $\mathcal{O}(k^2)$ rounds. For simplicity, we assume n is even and so $t = n/2 + k$. (Everything that follows works also for n odd, though things can be optimized somewhat.) Set $c \stackrel{\text{def}}{=} 2k$; this is equal to the difference between the number of dishonest parties and the number of honest parties. Without loss of generality, let P_1 be the sender. `Rand-Bcast` consists of two phases: Phase I takes exactly $\mathcal{O}(c^2)$ rounds, while Phase II runs for $\mathcal{O}(1)$ rounds in expectation. At the end of Phase I, each party in $\mathsf{Init} \stackrel{\text{def}}{=} \{P_1, \ldots, P_{c+1}\}$ outputs a message, which will be its final output for the entire protocol, while each party P_i in $\mathsf{Rem} \stackrel{\text{def}}{=} \{P_{c+2}, \ldots, P_n\}$ outputs a tuple of the form $\{(m_{i,1}, g_{i,1}), (m_{i,2}, g_{i,2}), \ldots, (m_{i,c+1}, g_{i,c+1})\}$. In the second phase, parties in $\mathsf{Rem} = \{P_{c+2}, \ldots, P_n\}$ determine their outputs using the output they obtained in Phase I. Parties in $\mathsf{Init} = \{P_1, \ldots, P_{c+1}\}$ do not take part in Phase II.

Phase I is based on the authenticated broadcast protocol of Dolev and Strong [13] which tolerates any $t < n$ dishonest parties and has the property that, in each round, honest parties send the same message to all other parties. Roughly speaking, parties $P_1, \ldots, P_{c+1}$ will execute the Dolev-Strong protocol with the following twist: whenever a party P_i (in the Dolev-Strong protocol) is supposed to send a message to every other party in $\{P_1, \ldots, P_{c+1}\}$, party P_i instead *gradecasts* the message *to all n parties in the network* using protocol `M-Gradecast` from Section 2.2. This has the effect of allowing parties $P_{c+2}, \ldots, P_n$ to "monitor" the execution of the Dolev-Strong protocol being run by parties $P_1, \ldots, P_{c+1}$.

The Dolev-Strong protocol guarantees that broadcast is achieved among $P_1, \ldots, P_{c+1}$ at the end of Phase I. As mentioned earlier, each remaining party $P_i \in \{P_{c+2}, \ldots, P_n\}$ outputs $\{(m_{i,1}, g_{i,1}), (m_{i,2}, g_{i,2}), \ldots, (m_{i,c+1}, g_{i,c+1})\}$ based on the messages and grades it received in Phase I. Informally, $m_{i,k}$ is the message that P_i "believes" P_k will output, with $g_{i,k}$ indicating the level of "confidence" P_i has in this determination. In particular, if P_k is honest then $m_{i,k}$ will be equal to the message output by P_k and $g_{i,k}$ will be the maximum possible grade. Furthermore, based on the properties of `M-Gradecast`, a relaxed form of agreement is achieved among the remaining parties. Specifically, for any honest parties $P_i, P_j \in \{P_{c+2}, \ldots, P_n\}$ and $k \in \{1, \ldots, c+1\}$ we have:

- If $g_{i,k} > 1$, then $m_{i,k} = m_{j,k}$ and $g_{j,k} \geq g_{i,k} - 1$.
- If $g_{i,k} = 1$, then $m_{i,k} = m_{j,k}$ or $g_{j,k} = 0$.

Therefore, although the remaining honest parties may not reach a unanimous decision when P_k is dishonest, the remaining honest parties will only disagree by "a small amount".

In Phase II, each remaining party P_i first *locally* "combines" its output $\{(m_{i,1}, g_{i,1}), (m_{i,2}, g_{i,2}), \ldots, (m_{i,c+1}, g_{i,c+1})\}$ into a single message/grade pair (m_i, g_i), with $g_i \in \{0, 1, 2\}$, such that the following hold for all honest parties $P_i, P_j \in \{P_{c+2}, \ldots, P_n\}$:

- If there exists an honest party $P_k \in \{P_1, \ldots, P_{c+1}\}$, then m_i is equal to the message output by P_k, and $g_i = 2$ (the maximum possible grade).
- If $g_i = 2$, then $m_i = m_j$ and $g_j \geq 1$.

Finally, parties $P_{c+2}, \ldots, P_n$ determine their final output as in Phase II of the broadcast protocol for $t = n/2$ described earlier. The key observation is that if there exists even a single honest party $P_k \in \{P_1, \ldots, P_{c+1}\}$, then for every honest party $P_i \in \{P_{c+2}, \ldots, P_n\}$ it holds that $m_i = m_k$ (where m_k is the output of P_k) and $g_i = 2$; otherwise (i.e., if $P_1, \ldots, P_{c+1}$ are all dishonest), a majority of the remaining parties are honest, and so they can rely on the output of $\mathtt{BA}_{\text{HonestMaj}}$.

Gradecast is also used as a building block in the (expected) sub-linear broadcast protocols of [14, 23, 3, 21]. In these works, gradecast is used to replace the broadcast channel in various sub-protocols that are run among all n parties in the network; these sub-protocols achieve some relaxed functionality that suffices for achieving broadcast. Here, we use gradecast in a different way, by having a small *subset* of the parties run some sub-protocol while gradecasting their messages to all parties in the network.

We now describe the two phases of the protocol in more detail, and prove the protocol's correctness.

3.2.1 Phase I

Set $g^* \stackrel{\text{def}}{=} 2^{\lceil \log(c+1) \rceil + 1} + 2^{\lceil \log(c+1) \rceil} - 1$.[4] Recall that we assume, without loss of generality, that P_1 is the sender. Let $\mathsf{Init} \stackrel{\text{def}}{=} \{P_1, \ldots, P_{c+1}\}$ (these are the parties who run the Dolev-Strong protocol in the *initial* phase) and let $\mathsf{Rem} \stackrel{\text{def}}{=} \{P_{c+2}, \ldots, P_n\}$ (these are the parties who *remain* in the second phase). Each party $P_i \in \mathsf{Init} \setminus \{P_1\}$ has a variable M_i initialized to the empty set; each party $P_i \in \mathsf{Rem}$ has variables $g_{i,1}, \ldots, g_{i,c+1}$ all initialized to g^*, and variables $M_{i,1}, \ldots, M_{i,c+1}$ all initialized to the empty set.

Roughly speaking, when a party $P_i \in \mathsf{Init} \setminus \{P_1\}$ receives a new message that originated from P_1 (with correct signatures attached), then as long as $|M_i| < 2$ it signs and gradecasts the received message, and adds the message to M_i. However, P_i stops adding new messages once $|M_i| = 2$, as this means P_i has received valid signatures of the sender on two different messages (and so P_i knows the sender is dishonest). Each P_i determines its output based on the contents of M_i at the end of Phase I.

Each party $P_i \in \mathsf{Rem}$ acts as follows: every time it hears $P_j \in \mathsf{Init}$ gradecast a new message that originated from P_1 (with correct signatures attached), then as long as $|M_{i,j}| < 2$ it adds the message to $M_{i,j}$ and updates $g_{i,j}$ based on the grade it received in the aforementioned execution of gradecast. At the end of Phase I, P_i determines $M_{i,j}$ (i.e., its determination as to what P_j will output) based on the contents of $M_{i,j}$.

[4] Jumping ahead, the reason g^* is set to this particular value is related to the second phase of the protocol. In Phase II, the parties will combine $c+1$ message/grade pairs into a single message/grade pair in a sequence of $\log(c+1)$ steps. In each step, the maximum possible grade will be reduced by half, and we set g^* to this particular value so that the final grade will lie between 0 and 2.

Protocol `Rand-Bcast` — Phase I

Step 1: P_1 computes a signature σ of m, runs $\mathtt{M\text{-}Gradecast}((m, \sigma, P_1), g^*)$, outputs m, and exits the protocol.

Step j, for $2 \le j \le c+2$:

1. Each P_i does the following: For each gradecast performed in the previous step, let $(m'_{i,\ell}, g'_{i,\ell})$ be the local output (of party P_i) of an invocation of `M-Gradecast` with $P_\ell \in \mathsf{Init}$ as the sender. (Note: each P_ℓ may gradecast multiple times in a given step. The output of each gradecast is handled separately.) Let $m'_{i,\ell}$ have the form $(m, \sigma_{\alpha_0}, P_1, \sigma_{\alpha_1}, P_{\alpha_1}, \ldots, \sigma_{\alpha_{j-2}}, P_{\alpha_{j-2}} = P_\ell)$.

   If $P_1, P_{\alpha_1}, \ldots, P_{\alpha_{j-2}} \in \mathsf{Init}$ are all unique; σ_{α_0} is a valid signature on m by P_1; and σ_{α_k} is a valid signature on $\sigma_{\alpha_{k-1}}$ by P_{α_k} for $1 \le k \le j-2$ (if all these conditions hold, we say $m'_{i,\ell}$ is *valid in step j*), then:

   Case 1: $P_i \in \mathsf{Init} \setminus \{P_1\}$**.** If $j < c+2$, $m \notin M_i$ and $|M_i| < 2$, then: set $M_i := M_i \cup \{m\}$; compute a signature $\sigma_{\alpha_{j-1}}$ on $\sigma_{\alpha_{j-2}}$; and run $\mathtt{M\text{-}Gradecast}((m'_{i,\ell}, \sigma_{\alpha_{j-1}}, P_i), g^*)$.

   Case 2: $P_i \in \mathsf{Rem}$**.** Set $g_{i,\ell} := \min\{g_{i,\ell}, g'_{i,\ell}\}$. If $m \notin M_{i,\ell}$ and $|M_{i,\ell}| < 2$, then set $M_{i,\ell} := M_{i,\ell} \cup \{m\}$.

2. **If** $P_i \in \mathsf{Init} \setminus \{P_1\}$: Let $d \le 2$ denote the number of times P_i has already run `M-Gradecast` in this step. Run $2-d$ invocations of $\mathtt{M\text{-}Gradecast}(\text{'nothing'}, g^*)$. (This ensures that each $P_i \in \mathsf{Init} \setminus \{P_1\}$ acts as the sender in exactly two executions of `M-Gradecast` in each step.)

Output determination: Let $\perp$ and ϕ be two special symbols, with $\perp$ indicating that a party has received two different messages with valid signatures of the sender, and ϕ indicating that a party did not receive any messages with a valid signature of the sender.

Each party $P_i \in \mathsf{Init} \setminus \{P_1\}$ **does:** If $|M_i| = 2$, output $\perp$; if $|M_i| = 1$, output the message in M_i; if $|M_i| = 0$, output ϕ.

Each party $P_i \in \mathsf{Rem}$ **does:** For each $P_\ell \in \mathsf{Init}$, compute $m_{i,\ell}$ as follows:

- If $|M_{i,\ell}| = 2$, set $m_{i,\ell} := \perp$; if $|M_{i,\ell}| = 1$, set $m_{i,\ell}$ to be the message in $M_{i,\ell}$; if $|M_{i,\ell}| = 0$, set $m_{i,\ell} := \phi$.

The round complexity of Phase I is $\mathcal{O}(k^2)$ as claimed. We now state several properties related to the first phase of our protocol (proofs appear in the Appendix). Phase II of `Rand-Bcast` is described in Section 3.2.2.

Lemma 3. *If the sender P_1 is honest, the following holds at the end of Phase I:*

1. *All honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* m*;*

2. *For all honest parties* $P_i \in \mathsf{Rem}$*, it holds that* $m_{i,1} = m$ *and* $g_{i,1} = g^*$*. Furthermore, for each* $2 \leq j \leq c+1$ *it holds that* $m_{i,j} = m$ *or* $m_{i,j} = \phi$ *(this holds even if* P_j *is dishonest).*

The next three lemmas concern the case when there exists an honest party in $\mathsf{Init} \setminus \{P_1\}$.

Lemma 4. *If any honest party* $P_i \in \mathsf{Init} \setminus \{P_1\}$ *outputs* $\perp$*, then all honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* $\perp$*, and for any honest* $P_j \in \mathsf{Rem}$ *it holds that* $m_{j,i} = \perp$ *and* $g_{j,i} = g^*$ *at the end of Phase I.*

Lemma 5. *If any honest party* $P_i \in \mathsf{Init} \setminus \{P_1\}$ *outputs* ϕ*, then all honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* ϕ*, and for any honest* $P_j \in \mathsf{Rem}$ *it holds that* $m_{j,i} = \phi$ *and* $g_{j,i} = g^*$ *at the end of Phase I. Moreover, if* $m_{j,k} \neq \phi$ *for some* $k \in \{1, \ldots, c+1\}$*, then* $g_{j,k} \leq 1$*.*

Lemma 6. *If any honest party* $P_i \in \mathsf{Init} \setminus \{P_1\}$ *outputs* $m \notin \{\perp, \phi\}$*, then all honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* m*, and for any honest* $P_j \in \mathsf{Rem}$ *it holds that* $m_{j,i} = m$ *and* $g_{j,i} = g^*$ *at the end of Phase I. Moreover, if* $m_{j,k} \neq m$ *and* $m_{j,k} \neq \phi$ *for some* $k \in \{1, \ldots, c+1\}$*, then* $g_{j,k} \leq 1$*.*

The next lemma states that some relaxed form of agreement exists among the parties in Rem regarding their determination as to what a (dishonest) $P_\ell \in \mathsf{Init}$ outputs. (Note that the case of an honest P_ℓ is handled in the previous three lemmas.) The lemma follows directly from the properties of gradecast and the specification of Phase I.

Lemma 7. *For* $1 \leq \ell \leq c+1$*, at the end of Phase I:*

- *If an honest party* $P_i \in \mathsf{Rem}$ *has* $g_{i,\ell} > 1$*, then all honest parties* $P_j \in \mathsf{Rem}$ *have both* $m_{j,\ell} = m_{i,\ell}$ *and* $g_{j,\ell} \geq g_{i,\ell} - 1$*.*
- *If an honest party* $P_i \in \mathsf{Rem}$ *has* $g_{i,\ell} = 1$*, then all honest parties* $P_j \in \mathsf{Rem}$ *have either* $m_{j,\ell} = m_{i,\ell}$ *or* $g_{j,\ell} = 0$*.*

3.2.2 Phase II

In the second phase of the protocol, the parties in Rem determine their outputs based on the information they obtained in the first phase. Recall that by the end of Phase I, each P_i holds values $\{(m_{i,1}, g_{i,1}), (m_{i,2}, g_{i,2}), \ldots, (m_{i,c+1}, g_{i,c+1})\}$ where $0 \leq g_{i,j} \leq g^*$ for all $1 \leq j \leq c+1$. In Phase II, based on these values, each P_i first locally computes a single message/grade pair $(m_i^{(0)}, g_i^{(0)})$, and then determines its output as in Phase II of the protocol for $t = n/2$ described earlier. The message/grade $(m_i^{(0)}, g_i^{(0)})$ is computed from $\{(m_{i,1}, g_{i,1}), (m_{i,2}, g_{i,2}), \ldots, (m_{i,c+1}, g_{i,c+1})\}$ in a sequence of $\lceil \log(c+1) \rceil$ (non-interactive) steps: in each step the number of message/grade pairs is reduced by half by "combining" two adjacent message/grade pairs into a single pair.

Before we describe the second phase of the protocol, we first describe a subroutine which takes a value d, two messages m_1, m_2, and two grades g_1, g_2 (where $0 \leq g_1, g_2 \leq 2^{d+1} + 2^d - 1$) as input, and outputs a message m and a grade g (where $0 \leq g \leq 2^d + 2^{d-1} - 1$).

Subroutine $\texttt{Combine}(d, m_1, m_2, g_1, g_2)$

If $(m_1 = m_2)$ **then**
  $m := m_1$ and $g := \max\{g_1 - 2^d - 2^{d-1},\, g_2 - 2^d - 2^{d-1},\, 0\}$;
else if $(m_1 \neq m_2)$ and $(m_1 \neq \phi)$ and $(m_2 \neq \perp)$ **then**
  begin
  If $(g_1 \leq 1)$ and $(g_2 = 2^{d+1} + 2^d - 1)$ **then** $m := m_2$ and $g := 2^d + 2^{d-1} - 1$
  else if $(g_1 \leq 2)$ and $(g_2 \geq 2^{d+1} + 2^d - 2)$ **then** $m := m_2$ and $g := 2^d + 2^{d-1} - 2$
  ...
  else if $(g_1 \leq 2^d + 2^{d-1})$ and $(g_2 \geq 2^d + 2^{d-1})$ **then** $m := m_2$ and $g := 0$
  else $m := m_1$ and $g := \max\{g_1 - 2^d - 2^{d-1}, 0\}$
  end
else (Note: here, either ($m_1 = \phi$ and $m_2 \neq \phi$) or ($m_1 \neq \perp$ and $m_2 = \perp$))
  begin
  if $(g_2 \leq 1)$ and $(g_1 = 2^{d+1} + 2^d - 1)$ **then** $m := m_1$ and $g := 2^d + 2^{d-1} - 1$
  else if $(g_2 \leq 2)$ and $(g_1 \geq 2^{d+1} + 2^d - 2)$ **then** $m := m_1$ and $g := 2^d + 2^{d-1} - 2$
  ...
  else if $(g_2 \leq 2^d + 2^{d-1})$ and $(g_1 \geq 2^d + 2^{d-1})$ **then** $m := m_1$ and $g := 0$
  else $m := m_2$ and $g := \max\{g_2 - 2^d - 2^{d-1}, 0\}$
  end
output (m, g).

Each party invokes the above subroutine using as input its own set of message/grade pairs. Informally, if a "relaxed" form of agreement on the input message/grade pairs has been established among the parties, this "relaxed" form of agreement still holds for the output message/grade pair. We make three observations regarding $\texttt{Combine}$. The first observation states that if one of the input messages is equal to $\perp$ and the corresponding grade is the maximum grade possible, then the output message will be equal to $\perp$ and the output grade will be the maximum grade possible.

Observation 1. *If* $m_1 = \perp$ *(resp.,* $m_2 = \perp$*) and* $g_1 =$

$2^{d+1}+2^d-1$ *(resp.,* $g_2 = 2^{d+1}+2^d-1$*), then* $m = \perp$ *and* $g = 2^d + 2^{d-1} - 1$.

The second observation is that if one of the input messages is equal to $m' \notin \{\perp, \phi\}$, the corresponding grade is the maximum grade possible, and one of the three following conditions hold: (i) the other input message is equal to ϕ; (ii) the other input grade is "low" (i.e., at most 1); or (iii) the two input messages are the same, then the output message will be equal to m' and the output grade will be the maximum grade possible.

Observation 2. *If* $m_1 \notin \{\perp, \phi\}$*;* $g_1 = 2^{d+1}+2^d-1$*; and either (1)* $m_2 = \phi$ *or (2)* $g_2 \leq 1$ *or (3)* $m_2 = m_1$*, then* $m = m_1$ *and* $g = 2^d + 2^{d-1} - 1$*. Analogously, if* $m_2 \notin \{\perp, \phi\}$*;* $g_2 = 2^{d+1}+2^d-1$*; and either (1)* $m_1 = \phi$ *or (2)* $g_1 \leq 1$ *or (3)* $m_1 = m_2$*, then* $m = m_2$ *and* $g = 2^d + 2^{d-1} - 1$.

The third observation is that if one of the input messages is equal to ϕ, the corresponding grade is the maximum grade possible, and one of the two following conditions hold: (i) the other input message is equal to ϕ or (ii) the other input grade is "low" (i.e., at most 1), then the output message will be equal to ϕ and the output grade will be the maximum grade possible.

Observation 3. *If* $m_1 = \phi$*;* $g_1 = 2^{d+1}+2^d-1$*; and either (1)* $m_2 = \phi$ *or (2)* $g_2 \leq 1$*, then* $m = \phi$ *and* $g = 2^d+2^{d-1}-1$*. Analogously, if* $m_2 = \phi$*;* $g_2 = 2^{d+1}+2^d-1$*; and either (1)* $m_1 = \phi$ *or (2)* $g_1 \leq 1$*, then* $m = \phi$ *and* $g = 2^d + 2^{d-1} - 1$.

We are now ready to specify the second phase of the protocol. Recall that the parties in Init do *not* take part in this phase.

Protocol `Rand-Bcast` — Phase II:
Parties $P_i \in \mathsf{Rem}$ perform the following steps:

1. **For** $1 \leq j \leq c+1$ **set** $m_{i,j}^{(\lceil \log(c+1) \rceil)} := m_{i,j}$ and $g_{i,j}^{(\lceil \log(c+1) \rceil)} := g_{i,j}$
   for $c+2 \leq j \leq 2^{\lceil \log(c+1) \rceil}$ **set** $m_{i,j}^{(\lceil \log(c+1) \rceil)} := \phi$ and $g_{i,j}^{(\lceil \log(c+1) \rceil)} := 0$.
2. **For** $d := \lceil \log(c+1) \rceil$ **to** 1 **do**:
   for $e := 1$ **to** 2^{d-1} **do**: $(m_{i,e}^{(d-1)}, g_{i,e}^{(d-1)}) \leftarrow$ $\texttt{Combine}(d, m_{i,2e-1}^{(d)}, g_{i,2e-1}^{(d)}, m_{i,2e}^{(d)}, g_{i,2e}^{(d)})$.
3. Set $(m_i, g_i) := (m_{i,1}^{(0)}, g_{i,1}^{(0)})$.
   If $g_i = 2$ **then** P_i enters protocol $\texttt{BA}_{\text{HonestMaj}}$ with input m_i, terminates $\texttt{BA}_{\text{HonestMaj}}$ after K rounds (where K is the constant from Lemma 1), and outputs m_i.
   else (i.e., $g_i < 2$) P_i enters protocol $\texttt{BA}_{\text{HonestMaj}}$ with input m_i, runs $\texttt{BA}_{\text{HonestMaj}}$ until successful termination of the protocol, and outputs whatever directed to by $\texttt{BA}_{\text{HonestMaj}}$.

We prove the following technical lemma in the Apendix which states that relaxed agreement is established on the message/grade pairs $\{(m_i, g_i)\}$.

Lemma 8. *By the end of Phase II, the following holds for all honest parties* $P_i, P_j \in \mathsf{Rem}$*:*

- *If* $g_i > 1$*, then* $m_j = m_i$ *and* $g_j \geq g_i - 1$.
- *If* $g_i = 1$*, then* $m_j = m_i$ *or* $g_j = 0$.

We now argue that `Rand-Bcast` achieves broadcast. There are three cases:

The sender P_1 is honest. By Lemma 3, all honest parties in $\mathsf{Init} \setminus \{P_1\}$ output m. For any honest party $P_i \in \mathsf{Rem}$, it follows from Lemma 3 and Observation 2 that $m_i = m$ and $g_i = 2$ at the end of Phase II, which implies that P_i outputs m.

P_1 is dishonest but there is an honest party $P_i \in \mathsf{Init} \setminus \{P_1\}$. Suppose P_i outputs $\perp$. By Lemma 4, all honest parties in $\mathsf{Init} \setminus \{P_1\}$ output $\perp$. Lemma 4 and Observation 1 show that, at the end of Phase II, $m_j = \perp$ and $g_j = 2$ for any honest party $P_j \in \mathsf{Rem}$, which implies that P_j outputs $\perp$. On the other hand, if P_i outputs ϕ it follows from Lemma 5 and Observation 3 that all honest parties output ϕ. Finally, if P_i outputs $m \notin \{\perp, \phi\}$ it follows from Lemma 6 and Observation 2 that all honest parties output m.

All parties in Init are dishonest. This means that a strict majority of the parties in Rem are honest. There are two sub-cases. The first sub-case is that by the end of Phase II there exists an honest party $P_i \in \mathsf{Rem}$ such that $g_i = 2$. Then, by Lemma 8, $m_j = m_i$ for all honest parties P_j and so all honest parties will output the same value m_i. The second sub-case is that $g_i \leq 1$ for all honest parties P_i. In this case, it follows from the properties of $\texttt{BA}_{\text{HonestMaj}}$ that all honest parties output the same message.

Phase I terminates in exactly $\mathcal{O}(k^2)$ rounds. Arguing as in the case of $t = n/2$, we see that Phase II terminates in expected constant rounds. We thus obtain the following theorem:

Theorem 1. *There exists an authenticated randomized* n*-party broadcast protocol tolerating* $t = n/2 + k$ *dishonest parties that runs in (expected)* $\mathcal{O}(k^2)$ *rounds.*

4. A Lower Bound on the Round Complexity

We start by considering a group of k parties $P_1, P_2, \ldots, P_k$ such that only two of them are honest. We show that there does not exist any (randomized) broadcast protocol having any runs that terminate in fewer than $k-1$ rounds.

Consider a broadcast protocol Π for k parties that tolerates $k-2$ dishonest parties. For $1 \leq i \leq k$, we construct a protocol $\bar{\Pi}_i$ that is the same as Π except that:

- If $i = 1$, then P_1 ignores all the messages sent to it except for those from P_2, and only sends messages to P_2 (i.e., P_1 only communicates with P_2).
- If $2 \leq i \leq k-1$, P_i ignores all the messages sent to it except for those from P_{i-1} and P_{i+1}, and only sends messages to P_{i-1} and P_{i+1} (i.e., P_i only communicates with P_{i-1} and P_{i+1}).
- If $i = k$, then P_k ignores all the messages sent to it except for those from P_{k-1}, and only sends messages to P_{k-1} (i.e., P_k only communicates with P_{k-1}).

For $1 \leq i \leq k-1$ and $b \in \{0,1\}$, define scenario $S_i^{(b)}$ as follows:

- P_1 is the sender and the bit b is its input.
- All parties except for P_i and P_{i+1} are dishonest. The honest parties P_i and P_{i+1} execute the protocol Π; a dishonest party P_j executes the protocol $\bar{\Pi}_j$.

For any $2 \leq i \leq k$, party P_i cannot distinguish whether it is in $S_{i-1}^{(b)}$ or $S_i^{(b)}$. In scenario $S_1^{(b)}$, parties P_1 and P_2 are both honest. Thus, P_1 and P_2 have to output b by the end of the protocol. Since P_2 cannot distinguish whether it is in $S_1^{(b)}$ or $S_2^{(b)}$, we see that P_2 has to output b in scenario $S_2^{(b)}$ as well; this means that P_3 has to output b as well. Prior to round 1, however, the view of P_2 is completely independent of b, and so the view of P_3 is independent of b prior to round 2.

In general, in scenario $S_i^{(b)}$, parties P_i and P_{i+1} have to output b and the view of P_{i+1} is completely independent of b prior to round i. If b is chosen uniformly at random and Π terminates before round $k-1$, then in scenario $S_{k-1}^{(b)}$ the output of P_k will not be equal to b with probability at least $1/2$. Since Π is a broadcast protocol, Π cannot terminate before round $k-1$. We conclude that there does not exist any broadcast protocol that can terminate in less than $k-1$ rounds if $k-2$ out of k parties are dishonest.

Using standard player-partitioning techniques (see the Appendix), we can generalize the above to show:

Theorem 2. *There does not exist any (randomized) n-party broadcast protocol tolerating t dishonest parties that terminates in fewer than $2n/(n-t)-1$ rounds (when $n-t \geq 2$).*

References

[1] A. Bar-Noy, D. Dolev, C. Dwork, and H. R. Strong. Shifting gears: Changing algorithms on the fly to expedite Byzantine agreement. In *6th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 1987.

[2] M. Ben-Or. Another advantage of free choice: Completely asynchronous agreement protocols. In *2nd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 1983.

[3] M. Ben-Or, E. Pavlov, and V. Vaikuntanathan. Byzantine agreement in the full-information model in $O(\log n)$ rounds. In *38th Annual ACM Symposium on Theory of Computing (STOC)*, 2006.

[4] P. Berman and J. A. Garay. Efficient distributed consensus with $n = (3+\epsilon)t$ processors. In *5th Intl. Workshop on Distributed Algorithms (WDAG)*, 1991.

[5] P. Berman and J. A. Garay. Cloture votes: $n/4$-resilient, polynomial-time distributed consensus in $t+1$ rounds. *Mathematical Systems Theory*, 26(1):3–20, 1993.

[6] C. Cachin, K. Kursawe, and V. Shoup. Random oracles in Constantitnople: Practical asynchronous Byzantine agreement using cryptography. In *19th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 2000.

[7] B. Chor and C. Dwork. Randomization in Byzantine agreement. *Advances in Computing Research*, 4, 1989.

[8] R. Cleve. Limits on the security of coin flips when half the processors are faulty. In *18th Annual ACM Symposium on Theory of Computing (STOC)*, 1986.

[9] B. A. Coan. A communication-efficient canonical form for fault-tolerant distributed protocols. In *5th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 1986.

[10] J. Considine, M. Fitzi, M. Franklin, L. A. Levin, U. Maurer, and D. Metcalf. Byzantine agreement given partial broadcast. *J. Cryptology*, 18(3):191–217, 2005.

[11] D. Dolev. The Byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982.

[12] D. Dolev, R. Reischuk, and H. R. Strong. Early stopping in Byzantine agreement. *J. ACM*, 37(4):720–741, 1990.

[13] D. Dolev and H. Strong. Authenticated algorithms for Byzantine agreement. *SIAM J. Computing*, 12(4):656–666, 1983.

[14] P. Feldman and S. Micali. An optimal probabilistic protocol for synchronous Byzantine agreement. *SIAM J. Computing*, 26(4):873–933, 1997.

[15] M. Fischer, N. Lynch, and M. Paterson. Impossibility of distributed consensus with one faulty processor. *J. ACM*, 32(2):374–382, 1985.

[16] M. J. Fischer and N. A. Lynch. A lower bound for the time to assure interactive consistency. *Info. Proc. Lett.*, 14(4):183–186, 1982.

[17] M. Fitzi and J. A. Garay. Efficient player-optimal protocols for strong and differential consensus. In *22nd Annual ACM Symp. on Principles of Distributed Computing (PODC)*, 2003.

[18] J. A. Garay, P. D. MacKenzie, M. Prabhakaran, and K. Yang. Resource fairness and composability of cryptographic protocols. In *3rd Theory of Cryptography Conference (TCC)*, 2006.

[19] J. A. Garay and Y. Moses. Fully polynomial Byzantine agreement for $n > 3t$ processors in $t+1$ rounds. *SIAM J. Computing*, 27(1):247–290, 1998.

[20] S. Goldwasser and Y. Lindell. Secure multi-party computation without agreement. *J. Cryptology*, 18(3):247–287, 2005.

[21] S. Goldwasser, E. Pavlov, and V. Vaikuntanathan. Fault-tolerant distributed computing in full-information networks. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

[22] R. Gradwohl, S. P. Vadhan, and D. Zuckerman. Random selection with an adversarial majority. In *Advances in Cryptology — Crypto 2006*.

[23] J. Katz and C.-Y. Koo. On expected constant-round protocols for Byzantine agreement. In *Advances in Cryptology — Crypto 2006*.

[24] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Trans. Prog. Lang. Syst.*, 4(3):382–401, 1982.

[25] Y. Lindell, A. Lysyanskaya, and T. Rabin. On the composition of authenticated Byzantine agreement. In *34th Annual ACM Symposium on Theory of Computing (STOC)*, 2002.

[26] Y. Moses and O. Waarts. Coordinated traversal: $(t+1)$-round Byzantine agreement in polynomial time. *J. Algorithms*, 17(1):110–156, 1994.

[27] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing (STOC)*, 1989.

[28] J. Nielsen. A threshold pseudorandom function construction and its applications. In *Advances in Cryptology — Crypto 2002*.

[29] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[30] B. Pfitzmann and M. Waidner. Information-theoretic pseudosignatures and Byzantine agreement for $t \geq n/3$. Technical Report RZ 2882 (#90830), IBM Research, 1996.

[31] M. Rabin. Randomized Byzantine generals. In *24th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1983.

[32] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing (STOC)*, 1990.

[33] S. Toueg, K. J. Perry, and T. K. Srikanth. Fast distributed agreement. *SIAM J. Computing*, 16(3):445–457, 1987.

A. Deferred Proofs

A.1. Correctness of `M-Gradecast`

Lemma 2. *Protocol* `M-Gradecast`$(\cdot, g^*)$ *is a* g^**-gradecast protocol with round complexity* $2g^* + 1$.

Proof. We first prove correctness. Suppose the sender is honest and let P_i be any honest party. All parties receive (m, σ) in round 1. Since the adversary cannot forge signatures, $|S_i| = 1$ and $m_i = m$ at all times. Hence $\bar{g}_i = 2g^*$ by the end of the protocol and P_i will output (m, g^*).

Next we prove soundness. Suppose there exists an honest party P_i that outputs $g_i \geq 1$. Note that $\bar{g}_i \geq 2g_i$ by the end of the protocol. Let round r_1 be the round during which m_i is added to S_i by P_i. Then $|S_i| = 0$ (and hence $\bar{g}_i = 0$) prior to round r_1. We claim that if there exists an honest party P_j who receives (m', σ') in round r_2 such that $m' \neq m_i$ and σ' is a valid signature on m by the sender, then $r_2 > r_1 + 2g_i - 3$. Assume the claim is not true, i.e., $r_2 \leq r_1 + 2g_i - 3$. Since P_j is honest, it sends (m', σ') to all parties (including P_i) in round $r_2 + 1$. Then by the end of step (a) in round $r_2 + 2$, it holds that $|S_i| \geq 2$ (note that S_i contains m_i by then as m_i is the first message added to it). Hence the value of $\bar{g}_i$ is at most $r_2 + 2 - r_1 \leq 2g_i - 1$, a contradiction.

We now complete the proof. Since P_i is honest, P_i sends m_i along with a valid signature from the sender to all parties in round r_1. All honest parties receive it by the end of round r_1. The claim we proved in the last paragraph states that no honest party P_j receives a different message $m' \neq m_i$ (with a valid signature from the sender) in or before round $r_1 + 2g_i - 3$. Consider the value of $\bar{g}_j$ by the end of the protocol. If $g_i \geq 2$, then $\bar{g}_j \geq r_1 + 2g_i - 3 + 1 - r_1 \geq 2g_i - 2$, and so P_j outputs m_i with $g_j \geq g_i - 1$. For the case $g_i = 1$, following the claim in the previous paragraph, no honest party P_j receives a message m' different from m_i (with a valid signature from the sender) in or before round $r_1 - 1$. Since P_j receives m_i (along with a valid signature from the sender) in round r_1, it holds that $m_i \in S_j$ by the end of step (a) in round $r_1 + 1$. It follows that $g_j = 0$ (if a different message m' is received by P_j in round r_1) or $m_j = m_i$. $\square$

A.2. Properties of Rand-Bcast

Lemma 3. *If the sender P_1 is honest, the following holds at the end of Phase I:*

1. *All honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* m*;*
2. *For all honest parties* $P_i \in \mathsf{Rem}$*, it holds that* $m_{i,1} = m$ *and* $g_{i,1} = g^*$*. Furthermore, for each* $2 \leq j \leq c+1$ *it holds that* $m_{i,j} = m$ *or* $m_{i,j} = \phi$ *(this holds even if* P_j *is dishonest).*

Proof. If the sender P_1 is honest, then in step 1 all honest parties $P_i \in \mathsf{Init} \setminus \{P_1\}$ receive (m, σ, P_1) as the output of the gradecast by P_1, where σ is a valid signature on m by P_1. Hence $m \in M_i$ by the end of step 2. Since the adversary cannot forge a signature of P_1, no message besides m will be added to M_i by the end of Phase I. Thus, all honest parties $P_i \in \mathsf{Init} \setminus \{P_1\}$ will output m.

For all honest parties $P_j \in \mathsf{Rem}$, we have $m_{j,1} = m$ and $g_{j,1} = g^*$ by the properties of M-Gradecast. Furthermore, $m_{j,i} = m$ or $m_{j,i} = \phi$ for any $2 \leq i \leq c+1$ as the adversary cannot forge a valid signature of P_1. $\square$

We prove two technical results that will be used in the proofs of Lemmas 4– 6.

Lemma 9. *Let* $P_i \in \mathsf{Init} \setminus \{P_1\}$ *be honest. If* $m \in M_i$ *by the end of Phase I, then:*

1. *For any honest* $P_j \in \mathsf{Init} \setminus \{P_1\}$ *it holds that* $m \in M_j$ *or* $|M_j| = 2$.
2. *For any honest* $P_j \in \mathsf{Rem}$ *it holds that* $m \in M_{j,i}$.

Proof. Suppose m is added to M_i in step k. Then in step $k-1$, party P_i received a message $m'_{i,\alpha_{k-2}} = (m, \sigma_{\alpha_0}, P_1, \sigma_{\alpha_1}, P_{\alpha_1}, \ldots, \sigma_{\alpha_{k-2}}, P_{\alpha_{k-2}})$ as the output of a gradecast by some party $P_{\alpha_{k-2}}$. In step k, P_i verifies that $m'_{i,\alpha_{k-2}}$ is valid, adds m to M_i, computes a signature $\sigma_{\alpha_{j-1}}$ of $\sigma_{\alpha_{j-2}}$, and gradecasts $(m'_{\alpha_{k-2},i}, \sigma_{\alpha_{j-1}}, P_i)$. All honest parties receive $(m'_{i,\alpha_{k-2}}, \sigma_{\alpha_{j-1}}, P_i)$ as the output of that gradecast. Since m is added to M_i in step k, it means that m is not in M_i in step $k-1$. Therefore, $P_i \notin \{P_1, P_{\alpha_1}, \ldots, P_{\alpha_{k-2}}\}$. This implies that $k \leq c+1$.

We know that $(m'_{i,\alpha_{k-2}}, \sigma_{\alpha_{j-1}}, P_i)$ is valid in step $k+1$. Consider an honest $P_j \in \mathsf{Init} \setminus \{P_1\}$. If m is not added to M_j in step $k+1$, then it means that m is already in M_j or $|M_j| = 2$ by the end of step $k+1$. This proves the first item. Next consider an honest party $P_j \in \mathsf{Rem}$. Following the properties of M-Gradecast and the protocol description, $m \in M_{j,i}$ by the end of step $k+1$, which proves the second item. $\square$

Lemma 10. *Let* $P_i \in \mathsf{Rem}$ *be honest. If, for some* $P_j \in \mathsf{Init}$*, it holds that* $m \in M_{i,j}$ *and* $g_{i,j} \geq 2$ *at the end of Phase I, then for all honest parties* $P_k \in \mathsf{Init} \setminus \{P_1\}$ *it holds that either* $m \in M_k$ *or* $|M_k| = 2$ *at the end of Phase I.*

Proof. Suppose m is added to $M_{i,j}$ in step r. This means P_j gradecasts $m_j = (m, \sigma_{\alpha_0}, \ldots, \sigma_{\alpha_{r-2}}, P_j)$ in step $r-1$, and P_i receives m_j with grade at least 2. Following the properties of M-Gradecast, all honest parties receive m_j with grade at least 1. We know that m_j is valid in step r since m is added to $M_{i,j}$ in step r. Therefore, by the end of step r, it holds that $m \in M_k$ or $|M_k| = 2$ for all honest parties $P_k \in \mathsf{Init} \setminus \{P_1\}$. $\square$

Lemma 4. *If any honest party* $P_i \in \mathsf{Init} \setminus \{P_1\}$ *outputs* $\perp$*, then all honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* $\perp$*, and for any honest* $P_j \in \mathsf{Rem}$ *it holds that* $m_{j,i} = \perp$ *and* $g_{j,i} = g^*$ *at the end of Phase I.*

Proof. If P_i outputs $\perp$, then $|M_i| = 2$ by the end of Phase I. Using Lemma 9, by the end of Phase I $|M_j| = 2$ for all honest parties $P_j \in \mathsf{Init} \setminus \{P_1\}$. Therefore P_j outputs $\perp$. If $P_j \in \mathsf{Rem}$ is honest, P_j always receives grade g^* in every gradecast by P_i. By Lemma 9, $m_{j,i} = \perp$ and $g_{j,i} = g^*$. $\square$

We prove Lemma 6 first, since we rely on it to prove Lemma 5.

Lemma 6. *If any honest party* $P_i \in \mathsf{Init} \setminus \{P_1\}$ *outputs* $m \notin \{\perp, \phi\}$*, then all honest parties in* $\mathsf{Init} \setminus \{P_1\}$ *output* m*, and for any honest* $P_j \in \mathsf{Rem}$ *it holds that* $m_{j,i} = m$ *and* $g_{j,i} = g^*$ *at the end of Phase I. Moreover, if* $m_{j,k} \neq m$ *and* $m_{j,k} \neq \phi$ *for some* $k \in \{1, \ldots, c+1\}$*, then* $g_{j,k} \leq 1$.

Proof. By the end of Phase I, $m \in M_i$. Consider an honest party $P_j \in \mathsf{Init} \setminus \{P_1\}$. By Lemma 9, we have $m \in M_j$ by the end of Phase I. If P_j does not output m, then $|M_j| = 2$ which means P_j outputs $\perp$. By Lemma 4, P_i should output $\perp$ instead of m, a contradiction.

Next consider an honest party $P_j \in \mathsf{Rem}$. We know that $m_{j,i} = m$ and $g_{j,i} = g^*$ by the properties of M-Gradecast. Now suppose there exists a $1 \leq k \leq c+1$ such that $m_{j,k} \neq m$ and $m_{j,k} \neq \phi$. Then there exists $m' \neq m$ such that $m' \in M_{j,k}$ by the end of Phase I. By Lemma 10, this means $g_{j,k} \leq 1$ or $m' \in M_i$ or $|M_i| = 2$ by the end of Phase I. Since P_i outputs m, we have $M_i = \{m\}$ and this means $g_{j,k} \leq 1$. $\square$

Lemma 5. *If any honest party $P_i \in \mathsf{Init} \setminus \{P_1\}$ outputs ϕ, then all honest parties in $\mathsf{Init} \setminus \{P_1\}$ output ϕ, and for any honest $P_j \in \mathsf{Rem}$ it holds that $m_{j,i} = \phi$ and $g_{j,i} = g^*$ at the end of Phase I. Moreover, if $m_{j,k} \neq \phi$ for some $k \in \{1, \ldots, c+1\}$, then $g_{j,k} \leq 1$.*

Proof. Consider an honest party $P_j \in \mathsf{Init} \setminus \{P_1\}$. If P_j does not output ϕ then, using Lemma 4 and Lemma 6, P_i should output $\perp$ or m' instead, a contradiction.

Now consider an honest party $P_j \in \mathsf{Rem}$. Properties of `M-Gradecast` imply that $m_{j,i} = \phi$ and $g_{j,i} = g^*$. Suppose there exists a $1 \leq k \leq c+1$ such that $m_{j,k} \neq \phi$. Then there exists an $m' \in M_{j,k}$ by the end of Phase I. Following Lemma 10, $g_{j,k} \leq 1$ or $m' \in M_i$ or $|M_i| = 2$. Since P_i outputs ϕ, this implies that $g_{j,k} \leq 1$. □

Lemma 8. *By the end of Phase II, the following holds for all honest parties $P_i, P_j \in \mathsf{Rem}$:*

- *If $g_i > 1$, then $m_j = m_i$ and $g_j \geq g_i - 1$.*
- *If $g_i = 1$, then $m_j = m_i$ or $g_j = 0$.*

Proof. The lemma follows once we show that, by the end of Phase II, for any $0 \leq d \leq \lceil \log(c+1) \rceil$ and $1 \leq e \leq 2^d$:

- If $g_{i,e}^{(d)} > 1$ for some honest party $P_i \in \mathsf{Rem}$, then $m_{j,e}^{(d)} = m_{i,e}^{(d)}$ and $g_{j,e}^{(d)} \geq g_{i,e}^{(d)} - 1$ for any honest party $P_j \in \mathsf{Rem}$.
- If $g_{i,e}^{(d)} = 1$ for some honest party $P_i \in \mathsf{Rem}$, then either $m_{j,e}^{(d)} = m_{i,e}^{(d)}$ or $g_{j,e}^{(d)} = 0$ for any honest party $P_j \in \mathsf{Rem}$.

We prove the above by induction on d.

Base Case: The statement is true for $d = \lceil \log(c+1) \rceil$ and any e by Lemma 7.

Inductive Step: Assume the statement is true for $d = d'+1$ and $e = 2e'-1$ and $e = 2e'$. We show that the statement is true for $d = d'$ and $e = e'$. We have the following cases:

1. Suppose that for all honest parties $P_i, P_j \in \mathsf{Rem}$, we have $m_{i,2e'-1}^{(d'+1)} = m_{j,2e'-1}^{(d'+1)}$. Consider the two subcases:
   - $m_{i,2e'}^{(d'+1)} = m_{j,2e'}^{(d'+1)}$ for all honest parties P_i, P_j. Then following the protocol specification, the statement is true for $d = d'$ and $e = e'$.
   - $m_{i,2e'}^{(d'+1)} \neq m_{j,2e'}^{(d'+1)}$ for some honest parties P_i, P_j. This means $g_{k,2e'}^{(d'+1)} \leq 1$ for all honest parties P_k. Following the protocol specification, if $g_{k,2e'-1}^{(d'+1)} > 2^d + 2^{d-1}$, then $m_{k,e'}^{(d')} = m_{k,2e'-1}^{(d'+1)}$ and $g_{k,e'}^{(d')} = g_{k,2e'-1}^{(d'+1)} - 2^d - 2^{d-1}$, else $g_{k,e'}^{(d')} = 0$. Thus the statement is true for $d = d'$ and $e = e'$.
2. Next suppose that for all honest parties $P_i, P_j \in \mathsf{Rem}$, it holds that $m_{i,2e'}^{(d'+1)} = m_{j,2e'}^{(d'+1)}$. The proof of this case is analogous to the previous case.
3. Finally, consider the case where neither condition above holds. This means that $g_{i,2e'-1}^{(d'+1)} \leq 1$ and $g_{i,2e'}^{(d'+1)} \leq 1$ for all honest parties P_i. Following the protocol specification, $g_{i,e'}^{(d')} = 0$. Hence the statement holds.

□

A.3. The Lower Bound

Theorem 2. *There does not exist any (randomized) n-party broadcast protocol tolerating t dishonest parties that terminates in fewer than $2n/(n-t)-1$ rounds (when $n-t \geq 2$).*

Proof. Let $h = n - t$. We divide the parties into $k = n/(h/2)$ disjoint groups $G_1, \ldots, G_k$, each of size $h/2$. Consider a broadcast protocol Π for n parties that can tolerate t dishonest parties. For $1 \leq i \leq k$, we construct a protocol $\bar{\Pi}_i$ that is the same as Π except that

- If $i = 1$, then the parties in G_1 ignore all the messages sent to them except for those from the parties in $G_1 \cup G_2$, and only send messages to the parties in $G_1 \cup G_2$ (i.e., parties in G_1 only communicates with parties in $G_1 \cup G_2$).
- If $2 \leq i \leq k-1$, parties in G_i ignore all the messages sent to them except for those from the parties in $G_{i-1} \cup G_i \cup G_{i+1}$, and only send messages to parties in $G_{i-1} \cup G_i \cup G_{i+1}$ (i.e., parties in G_i only communicates with parties in $G_{i-1} \cup G_i \cup G_{i+1}$).
- If $i = k$, then the parties in G_k ignore all the messages sent to them except for those from the parties in $G_{k-1} \cup G_k$, and only send messages to the parties in $G_{k-1} \cup G_k$ (i.e., parties in G_k only communicates with parties in $G_{k-1} \cup G_k$).

For $1 \leq i \leq k-1$ and $b \in \{0, 1\}$, define scenario $S_i^{(b)}$ as follows:

- The sender is in G_1 and the bit b is its input.
- All parties except for the parties in $G_i \cup G_{i+1}$ are dishonest.
- The honest parties in $G_i \cup G_{i+1}$ execute protocol Π; each dishonest party in G_j executes protocol $\bar{\Pi}_j$.

The rest of the proof proceeds analogously to the discussion in Section 4. □