On the Round Complexity of Randomized Byzantine Agreement

Ran Cohen
Boston University, MA, USA
Northeastern University, Boston, MA, USA
rancohen@ccs.neu.edu

Iftach Haitner
School of Computer Science, Tel Aviv University, Israel
iftachh@cs.tau.ac.il

Nikolaos Makriyannis
Department of Computer Science, Technion, Haifa, Israel
n.makriyannis@gmail.com

Matan Orland
School of Computer Science, Tel Aviv University, Israel
matanorland@mail.tau.ac.il

Alex Samorodnitsky
School of Engineering and Computer Science, The Hebrew University of Jerusalem, Israel
salex@cs.huji.ac.il

Abstract

We prove lower bounds on the round complexity of *randomized* Byzantine agreement (BA) protocols, bounding the halting probability of such protocols after one and two rounds. In particular, we prove that:

1. BA protocols resilient against $n/3$ [resp., $n/4$] corruptions terminate (under attack) at the end of the first round with probability at most $o(1)$ [resp., $1/2 + o(1)$].
2. BA protocols resilient against $n/4$ corruptions terminate at the end of the second round with probability at most $1 - \Theta(1)$.
3. For a large class of protocols (including all BA protocols used in practice) and under a plausible combinatorial conjecture, BA protocols resilient against $n/3$ [resp., $n/4$] corruptions terminate at the end of the second round with probability at most $o(1)$ [resp., $1/2 + o(1)$].

The above bounds hold even when the parties use a trusted setup phase, e.g., a public-key infrastructure (PKI).

The third bound essentially matches the recent protocol of Micali (ITCS'17) that tolerates up to $n/3$ corruptions and terminates at the end of the third round with constant probability.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols

Keywords and phrases Byzantine agreement, lower bound, round complexity

Digital Object Identifier 10.4230/LIPIcs.DISC.2019.12

Related Version A full version of the paper is available at `https://arxiv.org/abs/1907.11329`.

Funding *Ran Cohen*: Research supported by the Northeastern University Cybersecurity and Privacy Institute Post-doctoral fellowship, IARPA under award 2019-19020700009 (ACHILLES), NSF grant TWC-1664445, NSF grant 1422965, and by the NSF MACS project. Some of this work was done while the author was a post-doc at Tel Aviv University, supported by ERC starting grant 638121.
Iftach Haitner: Member of the Check Point Institute for Information Security. Research supported by ERC starting grant 638121.
Nikolaos Makriyannis: Research supported by ERC starting grant 638121 and by ERC advanced grant 742754.
Matan Orland: Research supported by ERC starting grant 638121.
Alex Samorodnitsky: Research partially supported by ISF grant 1724/15.

33rd International Symposium on Distributed Computing (DISC 2019).
Editor: Jukka Suomela; Article No. 12; pp. 12:1–12:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements We would like to thank Rotem Oshman, Juan Garay, Ehud Friedgut, and Elchanan Mossel for very helpful discussions.

1 Introduction

Byzantine agreement (BA) [56, 43] is one of the most important problems in theoretical computer science. In a BA protocol, a set of n parties wish to jointly agree on one of the honest parties' input bits. The protocol is *t-resilient* if no set of t corrupted parties can collude and prevent the honest parties from completing this task. In the closely related problem of *broadcast*, all honest parties must agree on the message sent by a (potentially corrupted) sender. Byzantine agreement and broadcast are fundamental building blocks in distributed computing and cryptography, with applications in fault-tolerant distributed systems [14, 42], secure multiparty computation [60, 33, 7, 15], and more recently, blockchain protocols [16, 31, 55].

In this work, we consider the *synchronous* communication model, where the protocol proceeds in rounds. It is well known that in the plain model, without any trusted setup assumptions, BA and broadcast can be solved if and only if $t < n/3$ [56, 43, 26, 30]. Assuming the existence of digital signatures and a public-key infrastructure (PKI), BA can be solved in the honest-majority setting $t < n/2$, and broadcast under any number of corruptions $t < n$ [23]. Information-theoretic variants that remain secure against computationally unbounded adversaries exist using information-theoretic pseudo-signatures [57].

An important aspect of BA and broadcast protocols is their *round complexity*. Deterministic t-resilient protocols require at least $t + 1$ rounds [25, 23], which is a tight lower bound [23, 30]. The breakthrough results of Ben-Or [6] and Rabin [58] showed that this limitation can be circumvented using randomization. In particular, Rabin [58] used *random beacons* (common random coins that are secret-shared among the parties in a trusted setup phase) to construct a BA protocol resilient to $t < n/4$ corruptions. Rabin's protocol fails with probability 2^{-r} after r rounds, and requires *expected* constant number of rounds to reach agreement. This line of research culminated with the work of Feldman and Micali [24] who showed how to compute the common coins from scratch, yielding expected-constant-round BA protocol in the plain model, resilient to $t < n/3$ corruptions. Katz and Koo [40] gave an analogue result in the PKI-model for the honest-majority case. Recent results used trusted setup and cryptographic assumptions to establish a surprisingly small expected round complexity, namely 9 for $t < n/3$ [47] and 10 for $t < n/2$ [49, 2].

The expected-constant-round protocols mentioned above are guaranteed to terminate (with negligible error probability) within a poly-logarithmic number of rounds. The lower bounds on the guaranteed termination from [25, 23] were generalized by [18, 39], showing that any randomized r-round protocol must fail with probability at least $(c \cdot r)^{-r}$ for some constant c. However, to date there is no lower bound on the *expected* round complexity of randomized BA.

In this work, we tackle this question and show new lower bounds for randomized BA. To make the discussion more informative, we consider a more explicit definition that bounds the halting probability within a specific number of rounds. A lower bound based on such a definition readily implies a lower bound on the expected round complexity of the BA protocol.

1.1 The Model

We start with describing in more details the model in which our lower bounds are given. In the BA protocols considered in this work, the parties are communicating over a synchronous network of private and authenticated channels. Each party starts the protocol with an input bit and upon completion decides on an output bit. The protocol is t-resilient if when facing t colluding parties that attack the protocol it holds that: (1) all honest parties agree on the same output bit (*agreement*), and (2) if all honest parties start with the same input bit, then this is the common output bit (*validity*). The protocols might have a *trusted setup phase*: a trusted external party samples correlated values and distributes them between the parties. A setup phase is known to be essential for tolerating $t \geq n/3$ corruptions, and seems to be crucial for highly efficient protocols such as [47, 16, 49, 2, 1]. The trusted setup phase is typically implemented using (heavy) secure multiparty computation [10, 12], via a public-key infrastructure, or with a random oracle (that can be used to model proof of work) [54].

Locally consistent adversaries. The attacks presented in the paper require very limited capabilities from the corrupted parties (a limitation that makes our bounds stronger). Specifically, a corrupted party might (1) prematurely abort, and (2) send messages to different parties based on *differing* input bits and/or incoming messages from other corrupted parties. We emphasize that corrupted parties sample their random coins honestly (and use the same coins for all messages sent). In addition, they do not lie about messages received from honest parties.

Public-randomness protocols. In many randomized protocols, including all those used in practice, cryptography is merely used to provide *message authentication* – preventing a party from lying about the messages it received – and *verifiable randomness* – forcing the parties to toss their coins correctly. The description of such protocols can be greatly simplified if only security against locally consistent adversaries is required (in which corrupted parties do not lie about their coin tosses and their incoming messages from honest parties). This motivates the definition of *public-randomness* protocols, where each party publishes its local coin tosses for each round (the party's first message also contains its setup parameter, if such exists). Although our attacks apply to arbitrary BA protocols, we show even stronger lower bounds for public-randomness protocols.

We illustrate the simplicity of the model by considering the BA protocol of Micali [47]. In this protocol, the cryptographic tools, digital signatures and verifiable random functions (VRFs),[1] are used to allow the parties elect leaders and toss coins with probability 2/3 as follows: each party P_i in round r evaluates the VRF on the pair (i, r) and multicasts the result. The leader is set to be the party with the smallest VRF value, and the coin is set to be the least-significant bit of this value. Since these values are uniformly distributed κ-bit strings (κ is the security parameter), and there are at least $2n/3$ honest parties, the success probability is 2/3. (Indeed, with probability 1/3, the leader is corrupted, and can send its value only to a subset of the parties, creating disagreement.)

When considering locally consistent adversaries, Micali's protocol can be significantly simplified by having each party randomly sample and multicast a uniformly distributed κ-bit string (cryptographic tools and setup phase are no longer needed). Corrupted parties can still send their values to a subset of honest parties as before, but they cannot send different random values to different honest parties.

[1] A pseudorandom function that provides a non-interactively verifiable proof for the correctness of its output.

A similar simplification applies to other BA protocols that are based on leader election and coin tosses such as [24, 27, 40] (private channels are used for a leader-election subprotocol), [49, 2] (cryptography is used for coin-tossing and message-authentication), and [16, 1] (cryptography is used to elect a small committee per round).[2]

▶ **Proposition 1** (Malicious security to locally consistent public-randomness protocol, informal). *Each of the BA protocols of [24, 27, 40, 47, 16, 49, 2, 1] induces a public-randomness BA protocol secure against locally consistent adversaries, with the same parameters.*

A useful abstraction for protocol design. To complete the picture, we remark that security against locally consistent adversaries, which may seem somewhat weak at first sight, can be compiled using standard cryptographic techniques into security against arbitrary adversaries. This reduction becomes lossless, efficiency-wise and security-wise, when applied to public-randomness protocols. Thus, building public-randomness protocols secure against locally consistent adversaries is a useful abstraction for protocol designers that want to use what cryptography has to offer, but without being bothered with the technical details.

Connection to the full-information model. The public-randomness model can be viewed as a restricted form of the *full-information model* [17, 8, 32, 5, 9, 35, 38, 44, 41, 45]. In the latter model, the adversary is computationally unbounded and has complete access to all the information in the system, i.e., it can listen to all transmitted messages and view the internal states of honest parties (such an adversary is also called *intrusive* [17]). One of the motivations to study full-information protocols is to separate *randomization* from *cryptography* and see to what extent randomization alone can speed up Byzantine agreement. Bar-Joseph and Ben-Or [5] showed that any full-information BA protocol tolerating $t = \Theta(n)$ adaptive, fail-stop corruptions (i.e., the adversary can dynamically choose which parties to crash) runs for $\tilde{\Omega}(\sqrt{n})$ rounds. Goldwasser et al. [35] constructed an $O(\log n)$-round BA protocol tolerating $t = (1/3 - \varepsilon)n$ static, malicious corruptions, for an arbitrarily small constant $\varepsilon > 0$.

We chose to state our results in the public-randomness model for two reasons. First, our lower bounds readily extend to lower bounds in the full-information model (since we consider weaker adversarial capabilities, e.g., all our attacks are efficient). Second, when considering locally consistent adversaries, public-randomness captures essentially what efficient cryptography has to offer. Indeed, all protocol used in practice can be cast as public-randomness protocols tolerating locally consistent adversaries (Proposition 1) and every public-randomness protocol secure against locally consistent adversaries can be compiled, using cryptography, to malicious security in the standard model, where security relies on secret coins (see Theorem 6 below).

We note that it is known how to compile certain full-information protocols and "boost" their security from fail-stop into malicious; however, these compilers capture either deterministic protocols [36, 13, 52] or protocols with a non-uniform source of randomness (namely, an SV-source [59]) [35]. It is unclear whether these compilers can be extended to capture arbitrary protocols (this is in fact stated as an open question in [13, 35]). In addition, these compilers are designed to be information theoretic and not rely on cryptography; thus, they do not model highly efficient protocols used in practice.

[2] Unlike the aforementioned protocols that use "simple" preprocess and "light-weight" cryptographic tools, the protocol of Rabin [58] uses a heavy, per execution, setup phase (consisting of Shamir sharing of a random coin for every potential round) that we do not know how to cast as a public-randomness protocol.

1.2 Our Results

We present three lower bounds on the halting probability of randomized BA protocols. To keep the following introductory discussion simple, we will assume that both validity and agreement properties hold perfectly, without error.

First-round halting. Our first result bounds the halting probability after a single communication round. This is the simplest case since parties cannot inform each other about inconsistencies they encounter. Indeed, the established lower bound is quite strong, showing an exponentially small bound on the halting probability when $t \geq n/3$, and exponentially close to $1/2$ when $t \geq n/4$.

▶ **Theorem 2** (First-round halting, informal). *Let Π be an n-party BA protocol and let γ denote the halting probability after a single communication round facing a locally consistent, static, adversary corrupting t parties. Then,*

- *$t \geq n/3$ implies $\gamma \leq 2^{t-n}$ for arbitrary protocols, and $\gamma = 0$ for public-randomness protocols.*
- *$t \geq n/4$ implies $\gamma \leq 1/2+2^{t-n}$ for arbitrary protocols, and $\gamma \leq 1/2$ for public-randomness protocols.*

Note that the deterministic $(t+1)$-round, t-resilient BA protocol of Dolev and Strong [23] can be cast as a locally consistent public-randomness protocol (in the plain model).[3] Theorem 2 shows that for $n = 3$ and $t = 1$, this two-round BA protocol is essentially optimal and cannot be improved via randomization (at least without considering complex protocols that cannot be cast as public-randomness protocols).

Second-round halting for arbitrary protocols. Our second result considers the halting probability after two communication rounds. This is a much more challenging regime, as honest parties have time to detect inconsistencies in first-round messages. Our bound for arbitrary protocols in this case is weaker, and shows that when $t > n/4$, the halting probability is bounded away from 1.

▶ **Theorem 3** (Second-round halting, arbitrary protocols, informal). *Let Π be an n-party BA protocol and let γ denote the halting probability after two communication rounds facing a locally consistent, static, adversary corrupting $t = (1/4+\varepsilon)\cdot n$ parties. Then, $\gamma \leq 1-(\varepsilon/5)^2$.*

Second-round halting for public-randomness protocols. Theorem 3 bounds the second-round halting probability of arbitrary BA protocols away from one. For public-randomness protocol we achieve a much stronger bound. The attack requires *adaptive* corruptions (as opposed to *static* corruptions in the previous case) and is based on a combinatorial conjecture that is stated below.[4]

[3] When considering locally consistent adversaries, the impossibility of BA for $t \geq n/3$ does not apply.

[4] The attack holds even without assuming Conjecture 5 when considering *strongly adaptive* corruptions [34], in which an adversary sees all messages sent by honest parties in any given round and, based on the messages' content, decides whether to corrupt a party (and alter its message or sabotage its delivery) or not. Similarly, the conjecture is not required if each party is limited to tossing a single unbiased coin. These extensions are not formally proved in this paper.

▶ **Theorem 4** (Second-round halting, public-randomness protocols, informal). *Let Π be an n-party public-randomness BA protocol and let γ denote the halting probability after two communication rounds facing a locally consistent adversary adaptively corrupting t parties. Then, for sufficiently large n and assuming Conjecture 5 holds,*

- $t > n/3$ *implies* $\gamma = 0$.
- $t > n/4$ *implies* $\gamma \leq 1/2$.

Theorem 4 shows that for sufficiently large n, any public-randomness protocol tolerating $t > n/3$ locally consistent corruptions cannot halt in less than three rounds (unless Conjecture 5 is false). In particular, its expected round complexity must be at least three.

To understand the meaning of this result, recall the protocol of Micali [47]. As discussed above, this protocol can be cast as a public-randomness protocol tolerating $t < n/3$ adaptive locally consistent corruptions. The protocol proceeds by continuously running a three-round sub-protocol until halting, where each sub-protocol consists of a coin-tossing round, a check-halting-on-0 round, and a check-halting-on-1 round. Executing a single instance of this sub-protocol demonstrates a halting probability of 1/3 after three rounds. By Theorem 4, a protocol that tolerates slightly more corruptions, i.e., $(1/3 + \varepsilon) \cdot n$, for arbitrarily small $\varepsilon > 0$, cannot halt in fewer rounds.

Our techniques. Our attacks follow the spirit of many lower bounds on the round complexity on BA and broadcast [25, 23, 39, 22, 29, 4]. The underlying idea is to start with a configuration in which validity assures the common output is 0, and gradually adjust it, while retaining the same output value, into a configuration in which validity assures the common output is 1. (For the simple case of deterministic protocols, each step of the argument requires the corrupted parties to lie about their input bits and incoming messages from other corrupted parties, but otherwise behave honestly.) Our main contribution, which departs from the aforementioned paradigm, is adding another dimension to the attack by aborting a random subset of parties (rather than simply manipulating the input and incoming messages). This change allows us to bypass a seemingly inherent barrier for this approach. We refer the reader to Section 2 for a detailed overview of our attacks.

We remark that a similar approach was employed by Attiya and Censor [3] for obtaining lower bounds on consensus protocols in the asynchronous shared-memory model, a flavor of BA in a communication model very different to the one considered in the present paper. Specifically, [3] showed that in an asynchronous shared-memory system, $\Theta(n^2)$ steps are required for n processors to reach agreement when facing $\Theta(n)$ *computationally unbounded strongly adaptive* corruptions (see Footnote 4). Their adversary also aborts a subset of the parties to prevent halting; however, the difference in communication model (synchronous in our work, vs. asynchronous in [3]) and the adversary's power (efficient and adaptive in our work, vs. computationally unbounded and strongly adaptive in [3]) yields a very different attack and analysis (though, interestingly, both attacks boil down to different variants of isoperimetric-type inequalities).

The combinatorial conjecture. We conclude the present section by motivating and stating the combinatorial conjecture assumed in Theorem 4, and discussing its plausibility. We believe the conjecture to be of independent interest, as it relates to topics from Boolean functions analysis such as influences of subsets of variables [53] and isoperimetric-type inequalities [50, 51]. The nature of our conjecture makes the following paragraphs somewhat technical, and reading them can be postponed until after going over the description of our attack in Section 2.

The analysis of our attack naturally gives rise to an isoperimetric-type inequality. For limited types of protocols, we manage to prove it using Friedgut's theorem [28] about approximate juntas and the KKL theorem [37]. For arbitrary protocols, however, we only manage to reduce our attack to the conjecture below.

We require the following notation before stating the conjecture. Let Σ denote some finite set. For $\boldsymbol{x} \in \Sigma^n$ and $\mathcal{S} \subseteq [n]$, define the vector $\perp_{\mathcal{S}}(\boldsymbol{x}) \in \{\Sigma \cup \perp\}^n$ by assigning all entries indexed by $\mathcal{S}$ with the value $\perp$, and all other entries according to $\boldsymbol{x}$. Finally, let $\mathbf{D}_{n,\sigma}$ denote the distribution induced over subsets of $[n]$ by choosing each element with probability σ independently at random.

▶ **Conjecture 5.** *For any $\sigma, \lambda > 0$ there exists $\delta > 0$ such that the following holds for large enough $n \in \mathbb{N}$: let Σ be a finite alphabet, and let $\mathcal{A}_0, \mathcal{A}_1 \subseteq \{\Sigma \cup \perp\}^n$ be two sets such that for both $b \in \{0,1\}$:*

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\Pr_{\boldsymbol{r} \leftarrow \Sigma^n} [\boldsymbol{r}, \perp_{\mathcal{S}}(\boldsymbol{r}) \in \mathcal{A}_b] \geq \lambda \right] \geq 1 - \delta.$$

Then,

$$\Pr_{\substack{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma} \\ \boldsymbol{r} \leftarrow \Sigma^n}} [\forall b \in \{0,1\} \colon \{\boldsymbol{r}, \perp_{\mathcal{S}}(\boldsymbol{r})\} \cap \mathcal{A}_b \neq \emptyset] \geq \delta.$$

Consider two large sets $\mathcal{A}_0$ and $\mathcal{A}_1$ which are "stable" in the following sense: for both $b \in \{0,1\}$, with probability $1-\delta$ over $\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}$, it holds that both $\boldsymbol{r}$ and $\perp_{\mathcal{S}}(\boldsymbol{r})$ belong to $\mathcal{A}_b$, with probability at least λ over $\boldsymbol{r}$. Conjecture 5 stipulates that with high probability ($\geq \delta$), the vectors $\boldsymbol{r}$ and $\perp_{\mathcal{S}}(\boldsymbol{r})$ lie in opposite sets (i.e., one is in $\mathcal{A}_0$ and the other $\mathcal{A}_{1-b}$), for random $\boldsymbol{r}$ and $\mathcal{S}$. It is somewhat reminiscent of the following flavor of isoperimetric inequality: for any two large sets $\mathcal{B}_0$ and $\mathcal{B}_1$, taking a random element from $\mathcal{B}_0$ and resampling a few coordinates, yields an element in $\mathcal{B}_1$ with large probability. Less formally, one can "move" from one set to the other by manipulating a few coordinates [50, 51].

A few remarks are in order. First, it suffices for our purposes to show that δ is a noticeable (i.e., inverse polynomial) function of n, rather than independent of n.[5] We opted for the latter as it gives a stronger attack. Second, the conjecture holds for "natural" sets such as balls, i.e., $\mathcal{A}_0$ and $\mathcal{A}_1$ are balls centered around 0^n and 1^n of constant radius,[6] and "prefix" sets, i.e., sets of the form $\mathcal{A}_b = b^k \times \{\Sigma \cup \perp\}^{n-k}$. Furthermore, the claim can be proven when the probabilities over $\mathcal{S}$ and $\boldsymbol{r}$ are reversed, i.e., "with probability λ over $\boldsymbol{r}$, it holds that both $\boldsymbol{r}$ and $\perp_{\mathcal{S}}(\boldsymbol{r})$ belong to $\mathcal{A}_b$ with probability at least $1-\delta$ over $\mathcal{S}$", instead of the above. Interestingly, this weaker statement boils down to the aforementioned isoperimetric-type inequality (c.f. [50] for the Boolean case and [51] for the non Boolean case).

We conclude by pointing out that, as mentioned in Footnote 4, the conjecture is not needed for certain limited cases that are not addressed in detail in the present paper. One such case is sketched out in Section 2.

1.3 Locally Consistent Security to Malicious Security

As briefly mentioned in Section 1.1, protocols that are secure against locally consistent adversaries can be compiled to tolerate arbitrary malicious adversaries. The compiler requires a PKI for digital signatures and verifiable random functions (VRFs) [48]. A VRF is a pseudorandom function with an additional property: using the secret key and an input x,

[5] We remark that it is rather easy to show that $\delta \geq 2^{-n}$, which is not good enough for our purposes.

[6] The alphabet Σ is not necessarily Boolean, and there are a couple of subtleties in defining balls.

the VRF outputs a pseudorandom value y along with a proof string π; using the public key, everyone can use π to verify whether y is the output of x. We consider a trusted setup phase for establishing the PKI, where every party generates keys for a VRF and for a signature scheme, and publishes the corresponding public keys.

Given a protocol that is secure against locally consistent adversaries, the compiled protocol proceeds as follows, round by round. Each party P_i sets its random coins for the r'th round ρ_i^r (together with a proof π_i^r) by evaluating the VRF over the pair (i, r). Next, for every $j \in [n]$, party P_i uses these coins to compute the message $m_{i \to j}^r$ for P_j, signs $m_{i \to j}^r$ along with the VRF proof π_i^r as $\sigma_{i \to j}^r$, and sends $(m_{i \to j}^r, \pi_i^r, \sigma_{i \to j}^r)$ to P_j. Finally, P_i proves to each P_j using a zero-knowledge proof of knowledge that:

1. There exist an input bit b, random coins ρ_i^r, as well as random coins and incoming messages $\rho_i^{r'}$ and $(m_{1 \to i}^{r'}, \ldots, m_{n \to i}^{r'})$ for every $r' < r$, such that: (1) π_i^r verifies that ρ_i^r is the VRF output of (i, r) (using the VRF public key of P_i), and (2) the message $m_{i \to j}^r$ is the output of the next-message function of P_i when applied to these values.
2. For every $r' < r$, the input bit b and the random coins $\rho_i^{r''}$ and incoming messages $(m_{1 \to i}^{r''}, \ldots, m_{n \to i}^{r''})$ for every $r'' < r'$, are the same as those used to generate $m_{i \to j}^{r'}$.
3. For $r > 1$, the messages received in the previous round are properly signed. That is, for every $k \in [n]$, there is a signature $\sigma_{k \to i}^{r-1}$ of the message $m_{k \to i}^{r-1}$ that verifies under the signature-verification key of P_k.

When considering public-randomness protocols, the above compilation can be made much more efficient. Instead of proving in zero knowledge the consistency of each message, each party P_i concatenates to each message all of its incoming messages from the previous round. A receiver can now locally verify the coins used by P_i are the VRF output of (i, r) (as assured by the VRF), that the incoming messages are properly signed, and that the message is correctly generated from the internal state of P_i (which is now visible and verified).

▶ **Theorem 6** (Locally consistent to malicious security, folklore, informal). *Assume PKI for digital signatures and VRF. Then, a BA protocol secure against locally consistent adversaries can be compiled into a maliciously secure BA protocol with the same parameters, apart from a constant blowup in the round complexity (no blowup for public-randomness protocols).*

1.4 Additional Related Work

Following the work of Feldman and Micali [24] in the two-thirds majority setting, Katz and Koo [40] improved the expected round complexity to 23, and Micali [47] to 9. In the honest-majority setting, Fitzi and Garay [27] showed expected-constant-round protocol and Katz and Koo [40] expected 56 rounds. Micali and Vaikuntanathan [49] adjusted the technique from [47] to the honest-majority case. Abraham et al. [2] achieved expected 10 rounds assuming static corruptions and expected 16 rounds assuming adaptive corruptions. Abraham et al. [1] constructed an expected-constant-round protocol tolerating $(1/2 - \epsilon) \cdot n$ adaptive corruptions with sublinear communication complexity. In the dishonest-majority setting, Garay et al. [29] constructed a broadcast protocol with expected $O(k)$ rounds, tolerating $t < n/2 + k$ corruptions.

Attiya and Censor-Hillel [4] extended the results of Chor et al. [18] and of Karlin and Yao [39] on guaranteed termination of randomized BA protocols to the asynchronous setting, and provided a tight lower bound.

Randomized protocols with expected constant round complexity have *probabilistic termination*, which requires delicate care with respect to composition (i.e., their usage as subroutines by higher-level protocols). Parallel composition of randomized BA protocols was analyzed in [6, 27], sequential composition in [46], and universal composition in [20, 19].

1.5 Open Questions

Our attack on two-round halting of public-randomness protocols is based on Conjecture 5. In this work we prove special cases of this conjecture, but proving the general case remains an open challenge.

A different interesting direction is to bound the halting probability of protocols when $t < n/4$. It is not clear how to extend our attacks to this regime.

2 Technical Overview

In this section, we outline our techniques for proving our results; we refer the reader to the full version of the paper [21] for formal claims and complete proofs. We start with explaining our bound for first-round halting of arbitrary protocols (Theorem 2). We then move to second-round halting, starting with the weaker bound for arbitrary protocols (Theorem 3), and then move to the much stronger bound for public-randomness protocols (Theorem 4).

Notations. We use calligraphic letters to denote sets, uppercase for random variables, lowercase for values, boldface for vectors, and sans-serif (e.g., A) for algorithms (i.e., Turing Machines). For $n \in \mathbb{N}$, let $[n] = \{1, \cdots, n\}$ and $(n) = \{0, 1, \cdots, n\}$. Let $\operatorname{dist}(x, y)$ denote the hamming distance between x and y. For a set $\mathcal{S} \subseteq [n]$ let $\overline{\mathcal{S}} = [n] \setminus \mathcal{S}$. For a set $\mathcal{R} \subseteq \{0,1\}^n$, let $\mathcal{R}|_{\mathcal{S}} = \{\boldsymbol{x}_{\mathcal{S}} \in \{0,1\}^{|\mathcal{S}|} \text{ s.t. } \boldsymbol{x} \in \mathcal{R}\}$, i.e., $\mathcal{R}|_{\mathcal{S}}$ is the projection of $\mathcal{R}$ on the index-set $\mathcal{S}$.

Fix an n-party randomized BA protocol $\Pi = (\mathsf{P}_1, \ldots, \mathsf{P}_n)$. For presentation purposes, we assume that validity and agreement hold *perfectly*, and consider no setup parameters (in the subsequent sections, we remove these assumptions). Furthermore, we only address here the case where the security threshold is $t > n/3$. The case $t > n/4$ requires an additional generic step that we defer to the technical sections of the paper. We denote by $\Pi(\boldsymbol{v}; \boldsymbol{r})$ the output of an honest execution of Π on input $\boldsymbol{v} \in \{0,1\}^n$ and randomness $\boldsymbol{r}$ (each party P_i holds input v_i and randomness r_i). We let $\Pi(\boldsymbol{v})$ denote the resulting random variable determined by the parties' random coins, and we write $\Pi(\boldsymbol{v}) = b$ to denote the event that the parties output b in an honest execution of Π on input $\boldsymbol{v}$. All corrupt parties described below are locally consistent (see Section 1.1).

2.1 First-Round Halting

Assume the honest parties of Π halt at the end of the first round with probability $\gamma > 0$ when facing t corruptions (on every input). Our goal is to upperbound the value of γ. Our approach is inspired by the analogous lower-bound for deterministic protocols (cf., [25, 23]). Namely, we start with a configuration in which validity assures the common output is 0, and, while maintaining the same output, we gradually adjust it into a configuration in which validity assures the common output is 1, thus obtaining a contradiction. For randomized protocols, the challenge is to maintain the invariant of the output, even when the probability of halting is far from 1. We make the following observations:

$$\text{Almost pre-agreement:} \quad \operatorname{dist}(\boldsymbol{v}, b^n) \leq t \implies \Pi(\boldsymbol{v}) = b. \tag{1}$$

That is, in an honest execution of Π, if the parties almost start with preagreement, i.e., with at least $n - t$ of b's in the input vector, then the parties output b with probability 1. Equation 1 follows from *agreement* and *validity* by considering an adversary corrupting exactly those parties with input $v_i \neq b$, and otherwise not deviating from the protocol.

$$\text{Neighboring executions (N1):} \quad \mathrm{dist}(\boldsymbol{v}_0, \boldsymbol{v}_1) \le t \implies \Pr_{\boldsymbol{r}}\left[\Pi(\boldsymbol{v}_0; \boldsymbol{r}) = \Pi(\boldsymbol{v}_1; \boldsymbol{r})\right] \ge \gamma. \tag{2}$$

That is, for two input vectors that are at most t-far (i.e., the resiliency threshold), the probability that the executions on these vectors yield the same output when using the same randomness is bounded below by the halting probability. To see why Equation 2 holds, consider the following adversary corrupting subset $\mathcal{C}$, for $\mathcal{C}$ being the set of indices where $\boldsymbol{v}$ and $\boldsymbol{v}'$ disagree. For an arbitrary partition $\{\overline{\mathcal{C}}_0, \overline{\mathcal{C}}_1\}$ of $\overline{\mathcal{C}}$, the adversary instructs $\mathcal{C}$ to send messages according to $\boldsymbol{v}_0$ to $\overline{\mathcal{C}}_0$ and according to $\boldsymbol{v}_1$ to $\overline{\mathcal{C}}_1$, respectively. With probability at least γ, all parties halt at the first round, and, by perfect agreement, all parties compute the same output.[7] Since parties in $\overline{\mathcal{C}}_b$ cannot distinguish this execution from a halting execution of $\Pi(\boldsymbol{v}_b; \boldsymbol{r})$, Equation 2 follows.

We deduce that if there are more than $n/3$ corrupt parties, then the halting probability is 0; this follows by combining the two observations above for $\boldsymbol{v}_0 = 0^{n-t}1^t$ and $\boldsymbol{v}_1 = 0^t 1^{n-t}$. Namely, by Equation 1, it holds that $\Pr_{\boldsymbol{r}}\left[\Pi(\boldsymbol{v}_0; \boldsymbol{r}) = \Pi(\boldsymbol{v}_1; \boldsymbol{r})\right] = 0$. Thus, by Equation 2, $\gamma = 0$.

2.2 Second-Round Halting – Arbitrary Protocols

We proceed to explain our bound for second-round halting of arbitrary protocols. Assume the honest parties of Π halt at the end of the second round with probability $\gamma > 0$ when facing t corruptions (on every input). Let $t = (1/3 + \varepsilon) \cdot n$, for an arbitrary small constant $\varepsilon > 0$. In spirit, the attack follows the footsteps of the single-round case described above; we show that neighboring executions compute the same output with good enough probability (related to the halting probability), and lower-bound the latter using the *almost pre-agreement* observation. There is, however, a crucial difference between the first-round and second-round cases; the honest parties can use the second round to detect whether (some) parties are sending inconsistent messages. Thus, the second round of the protocol can be used to "catch-and-discard" parties that are pretending to have different inputs to different parties, and so our previous attack breaks down. (In the one-round case, we exploit the fact that the honest parties cannot verify the consistency of the messages they received.) Still, we show that there is a suitable variant of the attack that violates the agreement of any "too-good" scheme.

At a very high level, the idea for proving the *neighboring* property is to *gradually* increase the set of honest parties towards which the adversary behaves according to $\boldsymbol{v}_1$ (for the remainder it behaves according to $\boldsymbol{v}_0$, which is a decreasing set of parties). While the honest parties might identify the attacking parties and discard their messages, they should still agree on the output and halt at the conclusion of the second round with high probability. We exploit this fact to show that at the two extremes (where the adversary is merely playing honestly according to $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$, respectively), the honest parties behave essentially the same. Therefore, if at one extreme (for $\boldsymbol{v}_0$) the honest parties output b, it follows that they also output b at the other extreme (for $\boldsymbol{v}_1$), which proves the *neighboring* property for the second-round case.

[7] In the above, we have chosen to ignore a crucial subtlety. In an execution of the protocol, it may be the case that there is a suitable message (according to $\boldsymbol{v}_0$ or $\boldsymbol{v}_1$) to prevent halting, yet the adversary cannot determine which one to send. In further sections, we address this issue by taking a random partition of $\overline{\mathcal{C}}$ (rather than an arbitrary one). By doing so, we introduce an error-term of $1/2^{n-t}$ when we upper bound the halting probability γ.

We implement the above by augmenting the one-round attack as follows. In addition to corrupting a set of parties that feign different inputs to different parties, the adversary corrupts an extra set of parties that is inconsistent with regards to the messages it received from the first set of corrupted parties. To distinguish between the two sets of corrupted parties, the former (first) will be referred to as "pivot" parties (since they pivot their input) and will be denoted $\mathcal{P}$, and the latter will be referred to as "propagating" parties (since they carefully choose what message to propagate at the second round) and will be denoted $\mathcal{L}$. We emphasize that the propagating parties deviate from the protocol only at the second round and only with regards to the messages received by the pivot parties (not with regards to their input – as is the case for the pivot parties). In more detail, we partition $\overline{\mathcal{P}} = [n] \setminus \mathcal{P}$ into $\ell = \lceil 1/\varepsilon \rceil$ sets $\{\mathcal{L}_1, \ldots, \mathcal{L}_\ell\}$, and we show that, unless there exists i such that parties in $\mathcal{C} = \mathcal{P} \cup \mathcal{L}_i$ violate agreement (explained below), the following must hold for neighboring executions.

$$\begin{aligned}&\text{Neighbouring executions (N2):}\quad \operatorname{dist}(\boldsymbol{v}_0, \boldsymbol{v}_1) \le n/3 \implies \\ &\Pr[\Pi(\boldsymbol{v}_0) = b \text{ in two rounds}] \ge \Pr[\Pi(\boldsymbol{v}_1) = b \text{ in two rounds}] - 2(\ell+1)^2 \cdot (1-\gamma).\end{aligned} \tag{3}$$

That is, for two input vectors that are at most $n/3$–far, the difference in probability that two distinct executions (for each input vector) yield the same output within two rounds is roughly upper-bounded by the quantity $(1-\gamma)/\varepsilon^2$ (i.e., non-halting probability divided by ε^2). To see that Equation 3 holds true, fix $\boldsymbol{v}_0, \boldsymbol{v}_1 \in \{0,1\}^n$ of hamming distance at most $n/3$, and let $\mathcal{P}$ be the set of indices where $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ differ. Consider the following $\ell + 1$ distinct variants of Π, denoted $\{\Pi_0, \ldots, \Pi_\ell\}$; in protocol Π_i, parties in $\mathcal{P}$ send messages to $\mathcal{L}_1, \ldots, \mathcal{L}_i$ according to the input prescribed by $\boldsymbol{v}_1$ and to $\mathcal{L}_{i+1}, \ldots, \mathcal{L}_\ell$ according to the input prescribed by $\boldsymbol{v}_0$, respectively. All other parties follow the instructions of Π for input $\boldsymbol{v}_0$. We write $\Pi_i = b$ to denote the event that the parties not in $\mathcal{P}$ output b. Notice that the endpoint executions Π_0 and Π_ℓ are identical to honest executions with input $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$, respectively. Let Halt_i denote the event that the parties not in $\mathcal{P}$ halt at the second round in an execution of Π_i. We point out that $\Pr[\neg\mathsf{Halt}_i] \le (\ell+1)\cdot(1-\gamma)$, since otherwise the adversary corrupting $\mathcal{P}$ and running Π_i, for a random $i \in (\ell) := \{0, \ldots, \ell\}$, prevents halting with probability greater than $1-\gamma$. Next, we inductively show that

$$\Pr[\Pi_i = b \wedge \mathsf{Halt}_i] \ge \Pr[\Pi_0 = b \wedge \mathsf{Halt}_0] - 2i \cdot (\ell+1)\cdot(1-\gamma), \tag{4}$$

for every $i \in (\ell)$, which yields the desired expression for $i = \ell$. In pursuit of contradiction, assume Equation 4 does not hold, and let i denote the smallest index for which it does not hold (observe that $i \neq 0$, by definition). Notice that

$$\begin{aligned}&\Pr[(\Pi_{i-1} = b \wedge \mathsf{Halt}_{i-1}) \wedge (\Pi_i \neq b \wedge \mathsf{Halt}_i)]\\ &\quad \ge \Pr[\Pi_{i-1} = b \wedge \mathsf{Halt}_{i-1}] - \Pr[\Pi_i = b \vee \neg\mathsf{Halt}_i]\\ &\quad \ge \Pr[\Pi_{i-1} = b \wedge \mathsf{Halt}_{i-1}] - \Pr[\Pi_i = b \wedge \mathsf{Halt}_i] - \Pr[\neg\mathsf{Halt}_i]\\ &\quad > 2\cdot(\ell+1)\cdot(1-\gamma) - \Pr[\neg\mathsf{Halt}_i]\\ &\quad \ge (\ell+1)\cdot(1-\gamma) > 0.\end{aligned}$$

The second inequality follows from union bound and $A \vee \neg B \equiv (A \wedge B) \vee \neg B$, the third inequality is by induction hypothesis, and the last inequality by the bound $\Pr[\neg\mathsf{Halt}_i] \le (\ell+1)\cdot(1-\gamma)$.

It follows that an adversary corrupting $\mathcal{C} = \mathcal{P} \cup \mathcal{L}_i$ causes disagreement with non-zero probability by acting as follows: parties in $\mathcal{P}$ and $\mathcal{L}_i$ send messages according to Π_i and Π_{i-1} to $\overline{\mathcal{C}}_0$ and $\overline{\mathcal{C}}_1$, respectively, where $\{\overline{\mathcal{C}}_0, \overline{\mathcal{C}}_1\}$ is an arbitrary partition of $\overline{\mathcal{C}} = [n] \backslash \mathcal{P} \cup \mathcal{L}_i$. Since disagreement is ruled out by assumption, we deduce Equations 4 and 3. To conclude, we combine the *almost pre-agreement* property (Equation 1) with the *neighboring* property (Equation 3) with $\boldsymbol{v}_0 = 0^{n-t}1^t$, $\boldsymbol{v}_1 = 0^t 1^{n-t}$, and $b = 1$. Namely, $\Pr[\Pi(\boldsymbol{v}_0) = 1 \text{ in two rounds}] = 0$, by *almost pre-agreement* and $\Pr[\Pi(\boldsymbol{v}_1) = 1 \text{ in two rounds}] \geq \gamma$, by *almost pre-agreement* and *halting.* It follows that $0 \geq \gamma - 2(\ell+1)^2 \cdot (1-\gamma)$, by Equation 3, and thus $1 - \frac{1}{2(\ell+1)^2+1} \geq \gamma$, which yields the desired expression.

2.3 Second-Round Halting – Public-Randomness Protocols

In Section 2.2, we ruled out "very good" second-round halting for arbitrary protocols via an efficient locally consistent attack. Recall that if the halting probability is too good (probability almost one), then there is a somewhat simple attack that violates agreement and/or validity. In this subsection, we discuss ruling out *any* second-round halting, i.e., halting probability bounded away from zero, for public-randomness protocols.

We first explain why the attack – as is – does not rule out second-round halting. Suppose that at the first round the parties of Π send a deterministic function of their input, and at the second round they send the messages they received at the first round together with a uniform random bit. On input $\boldsymbol{v}$ and randomness $\boldsymbol{r}$, the parties are instructed *not* to halt at the second round (i.e., carry on beyond the second round until they reach agreement with validity) if a super-majority ($\geq n-t$) of the v_i's are in agreement and $\operatorname{maj}(r_1, \ldots, r_n) \neq \operatorname{maj}(v_1, \ldots, v_n)$, i.e., the majority of the random bits does not agree with the super-majority of the inputs. In all other cases, the parties are instructed to output $\operatorname{maj}(r_1, \ldots, r_n)$. It is not hard to see that this protocol will halt with probability 1/2, even in the presence of the previous locally consistent adversary (regardless of the choice of propagating parties $\mathcal{L}_i$). More generally, if the randomness uniquely determines the output, the protocol designer can ensure that halting does not result in disagreement, by partitioning the randomness appropriately, and thus foiling the previous attack.[8]

To overcome the above apparent obstacle, we introduce another dimension to our locally consistent attack; we instruct an extra set of corrupted parties to abort at the second round without sending their second-round messages. By utilizing aborting parties, the adversary can potentially decouple the output/halting from the parties' randomness and thus either prevent halting or cause disagreement. In Section 2.3.1, we explain how to rule out second-round halting for a rather unrealistic class of public-randomness protocol. What makes the class of protocols unrealistic is that we assume security holds against unbounded locally consistent adversaries, and the protocol prescribes only a single bit of randomness per party per round. That being said, this case illustrates nicely our attack, and it also makes an interesting connection to Boolean functions analysis (namely, the KKL theorem [37]). For general public-randomness protocols, we only know how to analyze the aforementioned attack assuming Conjecture 5, as explained in Section 2.3.2.

[8] In Section 2.2, halting was close to 1 and thus the randomness was necessarily ambiguous regarding the output.

2.3.1 "Superb" Single-Coin Protocols

A BA protocol Π is *t-superb* if agreement and validity hold perfectly against an adaptive *unbounded* locally consistent adversary corrupting at most t parties, i.e., the probability that such an adversary violates agreement or validity is 0. A public-randomness protocol is *single-coin*, if, at any given round, each party samples a single unbiased bit.

▶ **Theorem 7** (Second-round halting, superb single-coin protocols)**.** *For every $\varepsilon > 0$ there exists $c > 0$ such that the following holds for large enough n. For $t = (1/3 + \varepsilon) \cdot n$, let Π be a t-superb, single-coin, n-party public-randomness Byzantine agreement protocol and let γ denote the probability that the protocol halts in the second round under a locally consistent attack. Then, $\gamma \leq n^{-c}$.*

We assume for simplicity that the parties do not sample any randomness at the first round, and write $\boldsymbol{r} \in \{0,1\}^n$ for the vector of bits sampled by the parties at the second round, i.e., r_i is a uniform random bit sampled by P_i.

As discussed above, our attack uses an additional set of corrupted parties of size $\sigma \cdot n$, dubbed the "aborting" parties and denoted $\mathcal{S}$, that abort indiscriminately at the second round (the value of σ is set to $\lfloor \varepsilon/4 \rfloor$ and $\ell = 2 \cdot \lceil 1/\varepsilon \rceil$ to accommodate for the new set of corrupted parties, i.e., $|\mathcal{L}_i| \leq n \cdot \varepsilon/2$). In more detail, analogously to the previous analysis, we consider $(\ell + 1) \cdot \binom{n}{\sigma n}$ distinct variants of Π, denoted $\{\Pi_i^{\mathcal{S}}\}_{i,\mathcal{S}}$ and indexed by $i \in (\ell)$ and $\mathcal{S} \subseteq [n]$ of size σn, as follows. In protocol $\Pi_i^{\mathcal{S}}$, parties in $\mathcal{P}$ send messages to $\mathcal{L}_1, \ldots, \mathcal{L}_i$ according to the input prescribed by $\boldsymbol{v}_1$, and to $\mathcal{L}_{i+1}, \ldots, \mathcal{L}_\ell$ according to the input prescribed by $\boldsymbol{v}_0$ (recall that $\mathcal{P}$ is exactly those indices where $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ differ). Parties in $\mathcal{S}$ act according to $\mathcal{P}$ or $\mathcal{L}_j$, for the relevant j, except that they abort at the second round without sending their second-round messages. We write $\Pi_i^{\mathcal{S}}(\boldsymbol{r}) = b$ to denote the event that the parties not in $\mathcal{P} \cup \mathcal{S}$ output b, where the parties' second-round randomness is equal to $\boldsymbol{r}$. Let $\mathsf{Halt}_i^{\mathcal{S}}$ denote the event that all parties not in $\mathcal{P} \cup \mathcal{S}$ halt at the second round in an execution of $\Pi_i^{\mathcal{S}}$, and define $\mathcal{R}_i^{\mathcal{S}}(b) = \{\boldsymbol{r} \in \{0,1\}^n \text{ s.t. } \Pi_i^{\mathcal{S}}(\boldsymbol{r}) = b \wedge \mathsf{Halt}_i^{\mathcal{S}}\}$. The following holds:

Neighbouring executions (N2†): (5)

$$\forall \boldsymbol{v}_0, \boldsymbol{v}_1 \in \{0,1\}^n \text{ with } \operatorname{dist}(\boldsymbol{v}_0, \boldsymbol{v}_1) \leq n/3, \quad \forall b \in \{0,1\}, i \in [\ell] := \{1, \ldots, \ell\}:$$
$$\left(\forall \mathcal{S}\colon \Pr\left[\Pi_{i-1}^{\mathcal{S}} = b \wedge \mathsf{Halt}_{i-1}^{\mathcal{S}}\right] \geq \gamma/2\right) \implies \left(\forall \mathcal{S}\colon \Pr\left[\Pi_i^{\mathcal{S}} = b \wedge \mathsf{Halt}_i^{\mathcal{S}}\right] \geq \gamma/2\right).$$

In words, for both $b \in \{0,1\}$: if $\Pi_{i-1}^{\mathcal{S}} = b$ and halts in two rounds with large probability ($\geq \gamma/2$), for every $\mathcal{S}$, then $\Pi_i^{\mathcal{S}} = b$ and halts in two rounds with large probability, for every $\mathcal{S}$. Before proving Equation 5, we show how to use it to derive Theorem 7. We apply Equation 5 for $\boldsymbol{v}_0 = 0^{n-t}1^t$, $\boldsymbol{v}_1 = 0^t1^{n-t}$, $b = 0$, and $i = \ell$, in combination with the properties of *validity* and *almost pre-agreement* (Equation 1). Namely, by these properties, a random execution of Π on input $\boldsymbol{v}_0$ where the parties in $\mathcal{S}$ abort at the second round yields output 0 with probability at least $\gamma/2$, for every $\mathcal{S} \in \binom{[n]}{\sigma n}$. Therefore, by Equation 5, we deduce that a random execution of Π on input $\boldsymbol{v}_1$ where the parties in $\mathcal{S}$ abort at the second round yields output 0 with probability at least $\gamma/2$, for every $\mathcal{S} \in \binom{[n]}{\sigma n}$. The latter violates either *validity* or *almost pre-agreement* – contradiction. To conclude the proof of Theorem 7, we prove Equation 5 by using the following corollary of the seminal KKL theorem [37] from Bourgain et al. [11]. (Recall that $\mathcal{R}|_{\overline{\mathcal{S}}}$ is the projection of $\mathcal{R}$ on the index-set $\overline{\mathcal{S}}$.)

▶ **Lemma 8.** *For every $\sigma, \delta \in (0,1)$, there exists $c > 0$ s.t. the following holds for large enough n. Let $\mathcal{R} \subseteq \{0,1\}^n$ be s.t. $|\mathcal{R}|_{\overline{\mathcal{S}}}| \leq (1 - \delta) \cdot 2^{(1-\sigma)n}$, for every $\mathcal{S} \subseteq [n]$ of size σn. Then, $|\mathcal{R}| \leq n^{-c} \cdot 2^n$.*

Loosely speaking, Lemma 8 states that for a set $\mathcal{R} \subseteq \{0,1\}^n$, if the size of every projection on a constant fraction of indices is bounded away from one (in relative size), then the size of $\mathcal{R}$ is vanishingly small (again, in relative size).[9]

Going back to the proof, in pursuit of contradiction, let $i \geq 1$ denote the smallest index for which Equation 5 does not hold, and without loss of generality suppose $b = 0$, i.e., there exists $\mathcal{S}$ such that $|\mathcal{R}_i^{\mathcal{S}}(0)| < \gamma/2 \cdot 2^n$, and $|\mathcal{R}_{i-1}^{\mathcal{S}'}(0)| \geq \gamma/2 \cdot 2^n$, for every relevant $\mathcal{S}'$. We prove Equation 5 by proving Equations 6 and 7, which result in contradiction via Lemma 8.

$$\text{Halting:} \qquad |\mathcal{R}_i^{\mathcal{S}}(1)| \geq \gamma/2 \cdot 2^n \tag{6}$$

$$\text{Perfect agreement:} \qquad \forall \mathcal{S}': \quad |\mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}'}}| \leq (1-\gamma/2) \cdot 2^{(1-\sigma)n} \tag{7}$$

Equation 6 follows by the *halting* property of $\Pi_i^{\mathcal{S}}$, since the execution halts if and only if $\boldsymbol{r} \in \mathcal{R}_i^{\mathcal{S}}(1) \cup \mathcal{R}_i^{\mathcal{S}}(0)$, and, by assumption, $|\mathcal{R}_i^{\mathcal{S}}(0)| < \gamma/2 \cdot 2^n$. To conclude, we prove Equation 7 by observing that for every $\mathcal{S}'$ and $b \in \{0,1\}$, and every $\boldsymbol{r}$ and $\boldsymbol{r}'$, if $\boldsymbol{r} \in \mathcal{R}_{i-1}^{\mathcal{S}'}(0)$ and $\boldsymbol{r}|_{\overline{\mathcal{S}'}} = \boldsymbol{r}'|_{\overline{\mathcal{S}'}}$, then $\boldsymbol{r}' \in \mathcal{R}_{i-1}^{\mathcal{S}'}(0)$ (by definition), i.e., membership to $\mathcal{R}_{i-1}^{\mathcal{S}'}(0)$ does not depend on the indices of $\mathcal{S}'$. It follows that $|\mathcal{R}_{i-1}^{\mathcal{S}}(0)|_{\overline{\mathcal{S}'}}| \geq \gamma/2 \cdot 2^{(1-\sigma)n}$, for every $\mathcal{S}'$, and therefore $|\mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}'}}| \leq (1-\gamma/2) \cdot 2^{(1-\sigma)n}$, since the sets $\mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}'}}$ and $\mathcal{R}_{i-1}^{\mathcal{S}}(0)|_{\overline{\mathcal{S}'}}$ are non-intersecting for every $\mathcal{S}'$. Otherwise, if $\mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}'}} \cap \mathcal{R}_{i-1}^{\mathcal{S}}(0)|_{\overline{\mathcal{S}'}} \neq \emptyset$, then the following attack violates the superb quality of the protocol. Fix $\mathcal{S}'$ and $\boldsymbol{r}$ such that $\boldsymbol{r} \in \mathcal{R}_i^{\mathcal{S}}(1)$ and $\boldsymbol{r}|_{\overline{\mathcal{S}'}} \in \mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}'}} \cap \mathcal{R}_{i-1}^{\mathcal{S}}(0)|_{\overline{\mathcal{S}'}}$, and consider the attacker controlling $\mathcal{P}$, $\mathcal{L}_i$, $\mathcal{S}$, and $\mathcal{S}'$ that sends messages according to $\Pi_i^{\mathcal{S}}$ and $\Pi_{i-1}^{\mathcal{S}'}$ to $\overline{\mathcal{C}}_0$ and $\overline{\mathcal{C}}_1$, respectively, where $\{\overline{\mathcal{C}}_0, \overline{\mathcal{C}}_1\}$ is an arbitrary partition of $\overline{\mathcal{C}} = [n] \setminus \mathcal{P} \cup \mathcal{L}_i \cup \mathcal{S} \cup \mathcal{S}'$. It is not hard to see the attacker violates agreement, whenever the randomness lands on $\boldsymbol{r}$.

▶ Remark 9. For superb, single-coin, public-randomness protocol, repeated application of Equation 2 and Lemma 8 rules out second-round halting for arbitrary (constant) fraction of corrupted parties (and not only $n/3$ fraction).

2.3.2 General (Public-Randomness) Protocols

The analysis above crucially relies on the superb properties of the protocol. While it can be generalized for protocols with near-perfect statistical security and constant-bit randomness, we only manage to analyze the most general case (i.e., protocols with non-perfect computational security and arbitrary-size randomness) assuming Conjecture 5. Very roughly (and somewhat inaccurately), when applying the above attack on general public-randomness protocols, the following happens for some $\delta > 0$ and both values of $b \in \{0,1\}$: for $(1-\delta)$-fraction of possible aborting subsets $\mathcal{S}$, the probability that the honest parties halt in two rounds and output the same value b, whether parties in $\mathcal{S}$ all abort or not, is bounded below by the halting probability. Assuming Conjecture 5, it follows that with probability δ over the randomness and $\mathcal{S}$, the honest parties under the attack output opposite values depending whether the parties in $\mathcal{S}$ abort or not. We conclude that the agreement of the protocol is at most δ.

References

1 Ittai Abraham, T.-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication Complexity of Byzantine Agreement, Revisited. In *Proceedings of the 38th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 317–326, 2019.

[9] In the jargon of Boolean functions analysis, since every large set has a $o(n)$-size index-set of influence almost one, it follows that some projection on a constant fraction of indices is almost full.

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*, 2019.

3 Hagit Attiya and Keren Censor. Tight bounds for asynchronous randomized consensus. *Journal of the ACM*, 55(5):20:1–20:26, 2008.

4 Hagit Attiya and Keren Censor-Hillel. Lower Bounds for Randomized Consensus under a Weak Adversary. *SIAM Journal on Computing*, 39(8):3885–3904, 2010.

5 Ziv Bar-Joseph and Michael Ben-Or. A Tight Lower Bound for Randomized Synchronous Consensus. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 193–199, 1998.

6 Michael Ben-Or. Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols (Extended Abstract). In *Proceedings of the 2nd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 27–30, 1983.

7 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1988.

8 Michael Ben-Or and Nathan Linial. Collective Coin Flipping, Robust Voting Schemes and Minima of Banzhaf Values. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 408–416, 1985.

9 Michael Ben-Or, Elan Pavlov, and Vinod Vaikuntanathan. Byzantine agreement in the full-information model in o(log n) rounds. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 179–186, 2006.

10 Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs. In *IEEE Symposium on Security and Privacy*, pages 287–304, 2015.

11 Jean Bourgain, Jeff Kahn, and Gil Kalai. Influential coalitions for Boolean Functions. In *CoRR*, 2014. `arXiv:1409.3033`.

12 Sean Bowe, Ariel Gabizon, and Matthew D. Green. A Multi-party Protocol for Constructing the Public Parameters of the Pinocchio zk-SNARK. In *Financial Cryptography and Data Security FC*, pages 64–77, 2018.

13 Gabriel Bracha. An Asynchronou [(n-1)/3]-Resilient Consensus Protocol. In *Proceedings of the 3rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 154–162, 1984.

14 Miguel Castro and Barbara Liskov. Practical Byzantine Fault Tolerance. In *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 173–186, 1999.

15 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty Unconditionally Secure Protocols (Extended Abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 11–19, 1988.

16 Jing Chen and Silvio Micali. Algorand. In *CoRR*, 2016. `arXiv:1607.01341`.

17 Benny Chor and Brian A. Coan. A Simple and Efficient Randomized Byzantine Agreement Algorithm. In *Fourth Symposium on Reliability in Distributed Software and Database Systems, SRDS*, pages 98–106, 1984.

18 Benny Chor, Michael Merritt, and David B. Shmoys. Simple constant-time consensus protocols in realistic failure models. *Journal of the ACM*, 36(3):591–614, 1989.

19 Ran Cohen, Sandro Coretti, Juan Garay, and Vassilis Zikas. Round-Preserving Parallel Composition of Probabilistic-Termination Cryptographic Protocols. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 37:1–37:15, 2017.

20 Ran Cohen, Sandro Coretti, Juan A. Garay, and Vassilis Zikas. Probabilistic Termination and Composability of Cryptographic Protocols. In *Advances in Cryptology – CRYPTO 2016, part III*, pages 240–269, 2016.

21 Ran Cohen, Iftach Haitner, Nikolaos Makriyannis, Matan Orland, and Alex Samorodnitsky. On the Round Complexity of Randomized Byzantine Agreement. *CoRR*, abs/1907.11329, 2019.

22 Danny Dolev, Rüdiger Reischuk, and H. Raymond Strong. Early Stopping in Byzantine Agreement. *Journal of the ACM*, 37(4):720–741, 1990.

23 Danny Dolev and Raymond Strong. Authenticated Algorithms for Byzantine Agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

24 Pesech Feldman and Silvio Micali. An Optimal Probabilistic Protocol for Synchronous Byzantine Agreement. *SIAM Journal on Computing*, 26(4):873–933, 1997.

25 Michael J. Fischer and Nancy A. Lynch. A Lower Bound for the Time to Assure Interactive Consistency. *Information Processing Letters*, 14(4):183–186, 1982.

26 Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy Impossibility Proofs for Distributed Consensus Problems. In *Proceedings of the 23th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 59–70, 1985.

27 Matthias Fitzi and Juan A. Garay. Efficient player-optimal protocols for strong and differential consensus. In *Proceedings of the 22th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 211–220, 2003.

28 Ehud Friedgut. Boolean Functions With Low Average Sensitivity Depend On Few Coordinates. *Combinatorica*, 18(1):27–35, 1998.

29 Juan A. Garay, Jonathan Katz, Chiu-Yuen Koo, and Rafail Ostrovsky. Round Complexity of Authenticated Broadcast with a Dishonest Majority. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 658–668, 2007.

30 Juan A. Garay and Yoram Moses. Fully polynomial Byzantine agreement in t+1 rounds. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 31–41, 1993.

31 Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP)*, pages 51–68, 2017.

32 Oded Goldreich, Shafi Goldwasser, and Nathan Linial. Fault-Tolerant Computation in the Full Information Model. *SIAM Journal on Computing*, 27(2):506–544, 1998.

33 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 218–229, 1987.

34 Shafi Goldwasser, Yael Tauman Kalai, and Sunoo Park. Adaptively Secure Coin-Flipping, Revisited. In *Proceedings of the 42th International Colloquium on Automata, Languages, and Programming (ICALP), part II*, pages 663–674, 2015.

35 Shafi Goldwasser, Elan Pavlov, and Vinod Vaikuntanathan. Fault-Tolerant Distributed Computing in Full-Information Networks. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 15–26, 2006.

36 Vassos Hadzilacos. Connectivity Requirements for Byzantine Agreement under Restricted Types of Failures. *Distributed Computing*, 2(2):95–103, 1987.

37 Jeff Kahn, Gil Kalai, and Nathan Linial. The Influence of Variables on Boolean Functions (Extended Abstract). In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 68–80, 1988.

38 Bruce M. Kapron, David Kempe, Valerie King, Jared Saia, and Vishal Sanwalani. Fast asynchronous Byzantine agreement and leader election with full information. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1038–1047, 2008.

39 Anna R. Karlin and Andrew Chi-Chih Yao. Probabilistic lower bounds for Byzantine agreement and clock synchronization. Unpublished manuscript, 1986.

40 Jonathan Katz and Chiu-Yuen Koo. On Expected Constant-Round Protocols for Byzantine Agreement. In *Advances in Cryptology – CRYPTO 2006*, pages 445–462, 2006.

41 Valerie King and Jared Saia. Byzantine agreement in polynomial expected time: [extended abstract]. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 401–410, 2013.

42 John Kubiatowicz, David Bindel, Yan Chen, Steven E. Czerwinski, Patrick R. Eaton, Dennis Geels, Ramakrishna Gummadi, Sean C. Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Y. Zhao. OceanStore: An Architecture for Global-Scale Persistent Storage. In *ASPLOS-IX Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 190–201, 2000.

43 Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

44 Allison B. Lewko. The Contest between Simplicity and Efficiency in Asynchronous Byzantine Agreement. In *Proceedings of the 25th International Symposium on Distributed Computing (DISC)*, pages 348–362, 2011.

45 Allison B. Lewko and Mark Lewko. On the complexity of asynchronous agreement against powerful adversaries. In *Proceedings of the 32th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 280–289, 2013.

46 Yehuda Lindell, Anna Lysyanskaya, and Tal Rabin. On the Composition of Authenticated Byzantine Agreement. *Journal of the ACM*, 53(6):881–917, 2006.

47 Silvio Micali. Very Simple and Efficient Byzantine Agreement. In *Proceedings of the 8th Annual Innovations in Theoretical Computer Science (ITCS) conference*, pages 6:1–6:1, 2017.

48 Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable Random Functions. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 120–130, 1999.

49 Silvio Micali and Vinod Vaikuntanathan. Optimal and player-replaceable consensus with an honest majority. Unpublished manuscript, 2017.

50 Elchanan Mossel, Ryan O'Donnell, Oded Regev, Jeffrey E. Steif, and Benny Sudakov. Non-interactive correlation distillation, inhomogeneous Markov chains, and the reverse Bonami-Beckner inequality. *Israel Journal of Mathematics*, 154(1):299–336, 2006.

51 Elchanan Mossel, Krzysztof Oleszkiewicz, and Arnab Sen. On Reverse Hypercontractivity. *Geometric and Functional Analysis*, 23(3):1062–1097, 2013.

52 Gil Neiger and Sam Toueg. Automatically Increasing the Fault-Tolerance of Distributed Algorithms. *Journal of Algorithms*, 11(3):374–419, 1990.

53 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

54 Rafael Pass and Elaine Shi. Hybrid Consensus: Efficient Consensus in the Permissionless Model. In *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, pages 39:1–39:16, 2017.

55 Rafael Pass and Elaine Shi. Thunderella: Blockchains with Optimistic Instant Confirmation. In *Advances in Cryptology – EUROCRYPT 2018, part II*, pages 3–33, 2018.

56 Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching Agreement in the Presence of Faults. *Journal of the ACM*, 27(2):228–234, 1980.

57 Birgit Pfitzmann and Michael Waidner. Unconditional Byzantine Agreement for any Number of Faulty Processors. In *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 339–350, 1992.

58 Michael O. Rabin. Randomized Byzantine Generals. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 403–409, 1983.

59 Miklos Santha and Umesh V. Vazirani. Generating Quasi-Random Sequences from Slightly-Random Sources (Extended Abstract). In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 434–440, 1984.

60 Andrew Chi-Chih Yao. Protocols for Secure Computations (Extended Abstract). In *Proceedings of the 23th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 160–164, 1982.