Journal of
CRYPTOLOGY

Check for
updates

Research Article

On the Round Complexity of Randomized Byzantine Agreement[*]

Ran Cohen
Reichman University, Herzliya, Israel
cohenran@idc.ac.il

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

Nikolaos Makriyannis
Fireblocks, New York, USA
n.makriyannis@gmail.com

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

Alex Samorodnitsky
School of Engineering and Computer Science, The Hebrew University of Jerusalem, 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 a fraction of corruptions greater than $1/4$ 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 a fraction

of corruptions greater than 1/3 [resp., 1/4] 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.

Keywords. Byzantine agreement, Lower bound, Round complexity.

1. Introduction

Byzantine agreement (BA) [50,63] 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 [16,49], secure multiparty computation [8,17,36,69], and more recently, blockchain protocols [18,35,62].

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 $t < n/3$ [28,32,50,63]. 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 $t < n$ [24]. Information-theoretic variants that remain secure against computationally unbounded adversaries exist using information-theoretic pseudo-signatures [65].

An important aspect of BA and broadcast protocols is their *round complexity*. For deterministic t-resilient protocols, $t + 1$ rounds are known to be sufficient [24,32] and necessary [24,27]. The breakthrough results of Ben-Or [6] and Rabin [66] showed that this limitation can be circumvented using randomization. In particular, Rabin [66] 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. The failure probability of Rabin's protocol after r rounds is 2^{-r}, and the *expected* number of rounds to reach agreement is constant. This line of research culminated with the work of Feldman and Micali [26] 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 [47] 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 $t < n/3$ [54] and 10 for $t < n/2$ [2,55].

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 [24,27] were generalized by [20,46], showing that any randomized r-round protocol must fail with probability at least $(c \cdot r)^{-r}$ for some constant c; in particular, randomized agreement with sub-constant failure

probability cannot be achieved in *strictly* constant rounds. 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*), (2) if all honest parties start with the same input bit, then this is the common output bit (*validity*), and (3) the protocol eventually terminates (*termination*). The protocols might have a *trusted setup phase*: a trusted external party samples correlated values (or receives a value from each party) and distributes them among 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 [1,2,18,54,55]. The trusted setup phase is typically implemented using (heavy) secure multiparty computation [10,13], distributed key generation [34,64], via a public-key infrastructure (see [14] for a discussion on different flavors of PKI), or with a random oracle (that can be used to model proofs of work) [61].

Locally consistent adversaries. The attacks presented in this paper require very limited capabilities from the corrupted parties (a limitation that makes our bounds stronger). Specifically, a corrupted party can deviate from the protocol only by: (1) prematurely aborting, and (2) altering (possibly a multiple number of times) its input bit and/or incoming messages from corrupted parties (see Sect. 3.1.2 for a precise definition). 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 [54]. 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.

A similar simplification applies to other BA protocols that are based on leader election and coin tosses such as [26,29,47] (private channels are used for a leader-election sub-protocol), [2,55] (cryptography is used for coin-tossing and message-authentication), and [1,18] (cryptography is used to elect a small committee per round).[2]

Proposition 1.1. (Malicious security to locally consistent public-randomness protocol, informal) *Each of the BA protocols of* [1,2,18,26,29,47,54,55] *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. See more details in Sect. 1.3.

Connection to the full-information model. The public-randomness model can be viewed as a restricted form of the *full-information model* [5,7,9,19,37,39,45,48,51,52]. 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* [19]). 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. [39] constructed

---

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

[2]Unlike the aforementioned protocols that use "simple" preprocess and "light-weight" cryptographic tools, the protocol of Rabin [66] 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.

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 1.1) 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 1.6 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 [15,42,59] or protocols with a non-uniform source of randomness (namely, an SV-source [67]) [39]. It is unclear whether these compilers can be extended to capture arbitrary protocols (this is in fact stated as an open question in [15,39]). 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.

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. Throughout we consider $t < n/2$ (as otherwise Byzantine agreement cannot be achieved).

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 1.2. (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,*

- *$n/2 > t \geq n/3$ implies $\gamma \leq 2^{t-n}$ for arbitrary protocols, and $\gamma = 0$ for public-randomness protocols.*
- *$n/2 > t \geq n/4$ 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 [24] can be cast as a locally consistent public-randomness protocol (in the plain model).[3] Theorem 1.2 shows that for $n = 3$ and $t = 1$, this two-round BA protocol is essentially

---

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

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 1.3. (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, $\gamma \le 1 - (\varepsilon/5)^2$.*

Second-round halting for public-randomness protocols. Theorem 1.3 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]

Theorem 1.4. (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 1.5 holds,*

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

Theorem 1.4 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 1.5 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 [54]. 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 1.4, 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 [4,24,25,27,33,46]. The underlying idea is to start

---

[4]The attack holds even without assuming Conjecture 1.5 when considering *strongly adaptive* corruptions [40], 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.

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 Sect. 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 1.4, 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 [60] and isoperimetric-type inequalities [57,58]. 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 Sect. 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 [31] about approximate juntas and the KKL theorem [44]. For arbitrary protocols, however, we can only reduce our attack to the conjecture below.

We require the following notation before stating the conjecture. Let Σ denote some finite set. For $x \in \Sigma^n$ and $\mathcal{S} \subseteq [n]$, define the vector $\perp_{\mathcal{S}}(x) \in \{\Sigma \cup \perp\}^n$ by assigning all entries indexed by $\mathcal{S}$ with the value $\perp$, and all other entries according to 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.

Conjectures 1.5. *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* $b \in \{0, 1\}$*:*

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

Then,

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

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 r and $\perp_{\mathcal{S}}(r)$ belong to $\mathcal{A}_b$, with probability at least λ over r. Conjecture 1.5 stipulates that with high probability ($\geq \delta$), the vectors r and $\perp_{\mathcal{S}}(r)$ lie in opposite sets (i.e., one is in $\mathcal{A}_0$ and the other $\mathcal{A}_1$), for random 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 [57,58].

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 r are reversed, i.e., "with probability λ over r, it holds that both r and $\perp_{\mathcal{S}}(r)$ 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 (cf. [57] for the Boolean case and [58] 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 Sect. 2.

1.3. *Locally Consistent Security to Malicious Security*

As briefly mentioned in Sect. 1.1, protocols that are secure against locally consistent adversaries can be compiled to tolerate arbitrary malicious adversaries. The compiler requires a PKI setup for digital signatures, verifiable random functions (VRFs) [56], and non-interactive zero-knowledge proofs (NIZK) [11]. A VRF is a pseudorandom function with an additional property: using the secret key and an input x, 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 a trusted party generates VRF and signature keys for every party, securely gives the secret keys to each party, and publishes the public keys to all.

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).

---

[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.

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 $m_{i \to j}^r$ as $\sigma_{i \to j}^r$, and sends $(m_{i \to j}^r, \sigma_{i \to j}^r, \pi_i^r)$ to P_j. Finally, P_i sends to P_j a NIZK proof that:

1. There exist an input bit b, random coins ρ_i^r, as well as random coins $\rho_i^{r'}$ and incoming messages and $(m_{1 \to i}^{r'}, \ldots, m_{n \to i}^{r'})$ for every prior round $r' < r$, 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), (2) the message $m_{i \to j}^r$ was signed by P_i, and (3) 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 $r > 1$, the messages $(m_{k \to i}^{r'}, \sigma_{k \to i}^{r'}, \pi_k^{r'})$ received by P_i from every P_k in prior rounds are proven to be properly generated. That is, P_k provided a NIZK proof that explains how $m_{k \to i}^{r'}$ was generated using random coins computed via the VRF on (k, r') and on incoming messages that were signed by the senders.

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 1.6. (Locally consistent to malicious security, folklore, informal) *Assume PKI for digital signatures, VRF, and NIZK. Then, an expected-constant-round BA protocol secure against locally consistent adversaries can be compiled into a maliciously secure protocol with the same parameters.*

The proof of Theorem 1.6 can be found in Sect. A.

1.4. *Additional Related Work*

Following the work of Feldman and Micali [26] in the two-thirds majority setting, Katz and Koo [47] improved the expected round complexity to 23, and Micali [54] to 9. In the honest-majority setting, Fitzi and Garay [29] showed expected-constant-round protocol and Katz and Koo [47] expected 56 rounds. Micali and Vaikuntanathan [55] adjusted the technique from [54] 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. [33] constructed a broadcast protocol with expected $O(k^2)$ rounds, tolerating $t < n/2 + k$ corruptions, that was improved by Fitzi and Nielsen [30] to expected $O(k)$ rounds.

Attiya and Censor-Hillel [4] extended the results of Chor et al. [20] and of Karlin and Yao [46] 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 [6,29], sequential composition in [53], and universal composition in [21,22].

1.5. *Open Questions*

Our attack on two-round halting of public-randomness protocols is based on Conjecture 1.5. 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.

Paper Organization

In Sect. 2 we present a technical overview of our attacks. The formal model and the exact bounds are stated in Sect. 3. The proof of the first-round halting is given in Sect. 4, and for second-round halting in Sect. 5. The proof of Theorem 1.6 appears in Sect. A.

2. Our Techniques

In this section, we outline our techniques for proving our results. We start with explaining our bound for first-round halting of arbitrary protocols (Theorem 1.2). We then move to second-round halting, starting with the weaker bound for arbitrary protocols (Theorem 1.3), and then move to the much stronger bound for public-randomness protocols (Theorem 1.4).

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 $[n] = \{1, \ldots, n\}$ and $(n) = \{0, 1, \ldots, n\}$. Let $\text{dist}(x, y)$ denote the hamming distance between x and y. For a set $\mathcal{S} \subseteq [n]$ let $\overline{\mathcal{S}} = [n] \backslash \mathcal{S}$. For a set $\mathcal{R} \subseteq \{0, 1\}^n$, let $\mathcal{R}|_{\mathcal{S}} = \{x_{\mathcal{S}} \in \{0, 1\}^{|\mathcal{S}|}$ s.t. $x \in \mathcal{R}\}$, 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 n is divisible by 3, 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 $\Pi(v; r)$ the output of an honest execution of Π on input $v \in \{0, 1\}^n$ and randomness r (each party P_i holds input v_i and randomness r_i). We let $\Pi(v)$ denote the resulting random variable determined by the parties' random coins, and we write $\Pi(v) = b$ to denote the event that the parties output b in an honest execution of Π on input v. All corrupt parties described below are locally consistent (see Sect. 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 (see [24, 27]). 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:

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

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.

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

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 Eq. (2) holds, consider the following adversary corrupting subset $\mathcal{C}$, for $\mathcal{C}$ being the set of indices where $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ 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 $\Pi(\boldsymbol{v}_b; \boldsymbol{r})$, Eq. (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 $\boldsymbol{v}_0 = 0^{2n/3}1^{n/3}$ and $\boldsymbol{v}_1 = 0^{n/3}1^{2n/3}$. Namely, by Eq. (1), it holds that $\Pr_{\boldsymbol{r}}\left[\Pi(\boldsymbol{v}_0; \boldsymbol{r}) = \Pi(\boldsymbol{v}_1; \boldsymbol{r})\right] = 0$. Thus, by Eq. (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

---

[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.

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.

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.

Neighbouring executions (N2): $\mathrm{dist}(\boldsymbol{v}_0, \boldsymbol{v}_1) \leq n/3 \implies$

$$\Pr\left[\Pi(\boldsymbol{v}_0) = b \text{ in two rounds}\right] \geq \Pr\left[\Pi(\boldsymbol{v}_1) = b \text{ in two rounds}\right] - 2(\ell + 1)^2 \cdot (1 - \gamma).$$

(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 Eq. (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 $\Pr\left[\neg\mathsf{Halt}_i\right] \leq (\ell + 1) \cdot (1 - \gamma)$, 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

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

for every $i \in (\ell)$, which yields the desired expression for $i = \ell$. In pursuit of contradiction, assume Eq. (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

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

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 $\Pr\left[\neg\mathsf{Halt}_i\right] \leq (\ell + 1) \cdot (1 - \gamma)$.

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 $\overline{\mathcal{C}} = [n] \setminus \mathcal{P} \cup \mathcal{L}_i$. Since disagreement is ruled out by assumption, we deduce Eq. (4) and (3). To conclude, we combine the *almost pre-agreement* property (Eq. (1)) with the *neighboring* property (Eq. (3)) with $\boldsymbol{v}_0 = 0^{2n/3}1^{n/3}$, $\boldsymbol{v}_1 = 0^{n/3}1^{2n/3}$, and $b = 1$. Namely, $\Pr\left[\Pi(\boldsymbol{v}_0) = 1 \text{ in two rounds}\right] = 0$, by *almost pre-agreement* and $\Pr\left[\Pi(\boldsymbol{v}_1) = 1 \text{ in two rounds}\right] \geq \gamma$, by *almost pre-agreement* and *halting*. It follows that $0 \geq \gamma - 2(\ell + 1)^2 \cdot (1 - \gamma)$, by Eq. (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 Sect. 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 close to 1, 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 v and randomness 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 $\mathrm{maj}(r_1, \ldots, r_n) \neq \mathrm{maj}(v_1, \ldots, v_n)$, 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 $\mathrm{maj}(r_1, \ldots, r_n)$. 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, then the protocol designer ensures 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 Sect. 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 [44]). For general public-randomness protocols, we only know how to analyze the aforementioned attack assuming Conjecture 1.5, as explained in Sect. 2.3.2.

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 2.1. (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 $r \in \{0, 1\}^n$ 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.

---

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

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 $\varepsilon/4$ 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 $\mathcal{P}$ consists of 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 †):

$$\forall \boldsymbol{v}_0, \boldsymbol{v}_1 \in \{0, 1\}^n \text{ with } \mathrm{dist}(\boldsymbol{v}_0, \boldsymbol{v}_1) \leq n/3, \quad \forall b \in \{0, 1\}, i \in [\ell] := \{1, \ldots, \ell\}:$$

$$\left(\forall \mathcal{S}: \ \Pr\left[\Pi_{i-1}^{\mathcal{S}} = b \wedge \mathsf{Halt}_{i-1}^{\mathcal{S}} \right] \geq \gamma/2 \right)$$
$$\implies \left(\forall \mathcal{S}: \ \Pr\left[\Pi_i^{\mathcal{S}} = b \wedge \mathsf{Halt}_i^{\mathcal{S}} \right] \geq \gamma/2 \right). \tag{5}$$

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 Eq. (5), we show how to use it to derive Theorem 2.1. We apply Eq. (5) for $\boldsymbol{v}_0 = 0^{2n/3}1^{n/3}$, $\boldsymbol{v}_1 = 0^{n/3}1^{2n/3}$, $b = 0$, and $i = \ell$, in combination with the properties of *validity* and *almost pre-agreement* (Eq. (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 Eq. (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 2.1, we prove Eq. (5) by using the following corollary of the seminal KKL theorem [44] from Bourgain et al. [12]. (Recall that $\mathcal{R}|_{\overline{\mathcal{S}}}$ is the projection of $\mathcal{R}$ on the index-set $\overline{\mathcal{S}}$.)

Lemma 2.2. *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 2.2 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 Eq. (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 Eq. (5) by proving Eqs. (6) and (7), which result in contradiction via Lemma 2.2.

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

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

Eq. (6) follows by the *halting* property of $\Pi_i^{\mathcal{S}}$, since the execution halts if and only if $r \in \mathcal{R}_i^{\mathcal{S}}(1) \cup \mathcal{R}_i^{\mathcal{S}}(0)$, and, by assumption, $|\mathcal{R}_i^{\mathcal{S}}(0)| < \gamma/2 \cdot 2^n$. To conclude, we prove Eq. (7) by observing that for every $\mathcal{S}'$ and $b \in \{0, 1\}$, and every r and r', if $r \in \mathcal{R}_{i-1}^{\mathcal{S}'}(0)$ and $r|_{\overline{\mathcal{S}}'} = r'|_{\overline{\mathcal{S}}'}$, then $r' \in \mathcal{R}_{i-1}^{\mathcal{S}'}(0)$ (by definition), i.e., membership to $\mathcal{R}_{i-1}^{\mathcal{S}'}(0)$ does not depend on the indices of $\mathcal{S}'$. Therefore, if $r \in \mathcal{R}_i^{\mathcal{S}}(1)$ and $r|_{\overline{\mathcal{S}}'} \in \mathcal{R}_{i-1}^{\mathcal{S}'}(0)|_{\overline{\mathcal{S}}'}$, for some $\mathcal{S}'$ and r, then $r \in \mathcal{R}_{i-1}^{\mathcal{S}'}(0) \cap \mathcal{R}_i^{\mathcal{S}}(1)$ which gives rise to the following attack. The attacker controls $\mathcal{P}, \mathcal{L}_i, \mathcal{S}$, and $\mathcal{S}'$, and 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 r.

Finally, since $|\mathcal{R}_{i-1}^{\mathcal{S}'}(0)| \geq \gamma/2 \cdot 2^n$, we observe that $|\mathcal{R}_{i-1}^{\mathcal{S}'}(0)|_{\overline{\mathcal{S}}'}| \geq \gamma/2 \cdot 2^{(1-\sigma)n}$, and, since $\mathcal{R}_{i-1}^{\mathcal{S}'}(0)|_{\overline{\mathcal{S}}'}$ and $\mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}}'}$ are non-intersecting for every $\mathcal{S}'$, it follows that $|\mathcal{R}_i^{\mathcal{S}}(1)|_{\overline{\mathcal{S}}'}| \leq (1 - \gamma/2) \cdot 2^{(1-\sigma)n}$, which yields Eq. (7).

Remark 2.3.   For superb, single-coin, public-randomness protocol, repeated application of Eq. (2) and Lemma 2.2 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 1.5. 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 $\mathcal{S}$

---

[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.

all abort or not, is bounded below by the halting probability. Assuming Conjecture 1.5, 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. We refer the reader to Sect. 5.2 for the full details.

3. Our Lower Bounds

In this section, we formally state our lower bounds on the round complexity of Byzantine agreement protocols. The communication and adversarial models as well as the notion of Byzantine agreement protocols we consider are given in Sect. 3.1, and our bounds are formally stated in Sect. 3.2.

3.1. *The Model*

3.1.1. *Protocols*

All protocols considered in this paper are PPT (probabilistic polynomial time): the running time of every party is polynomial in the (common) security parameter (given as a unary string). We only consider Boolean-input Boolean-output protocols: apart from the common security parameter, all parties have a single input bit, and each of the honest parties outputs a single bit. For an n-party protocol Π, an input vector $v \in \{0, 1\}^n$ and randomness r, let $\Pi(v; r)$ denote the output vector of the parties in an (honest) execution with party P_i's input being v_i and randomness r_i. For a set of parties $\mathcal{P} \subseteq [n]$, we denote by $\Pi(v; r)_{\mathcal{P}}$ the output vector of the parties in $\mathcal{P}$.

The protocols we consider might have a *setup phase* in which before interaction starts a trusted party distributes (correlated) values between the parties. We only require the security to hold for a *single* use of the setup parameters, i.e., for a single instance of the BA protocol (in reality, these parameters are set once and then used for many interactions). This, however, only makes our lower bound stronger.

The communication model is *synchronous*, meaning that the protocols proceed in rounds. In each round every party can send a message to every other party over a private and authenticated channel. (Allowing the protocol to be executed over private channels makes our lower bounds stronger.) It is guaranteed that all of the messages that are sent in a round will arrive at their destinations by the end of that round.

3.1.2. *Adversarial Model*

We consider both adaptive and non-adaptive (also known as, static) adversaries. An adaptive adversary can choose which parties to corrupt for the next round immediately after the conclusion of the previous round but before seeing the next round's messages. If a party has been corrupted then it is considered corrupt for the rest of the execution. A non-adaptive (static) adversary chooses which parties to corrupt *before* the execution of the protocol begins (i.e., before the setup phase, if such exists). We measure the success probability of the latter adversaries as the expectation over their choice of corrupted parties.

We consider both *rushing* and *non-rushing* adversaries. A non-rushing adversary chooses the corrupted parties' messages in a given round based on the messages sent in the *previous* rounds. In contrast, a rushing adversary can base the corrupted parties' messages on the messages sent in the previous rounds, and on those sent by the honest parties in the *current* round.

Locally consistent adversaries. As discussed in Sect. 1.1, our attack requires very limited capabilities from each corrupted party: to prematurely abort, and to lie about its input bit and incoming messages from other corrupted parties. In particular, a corrupted party tosses its local coins honestly and does not lie about incoming messages from honest parties. We now present the formal definition.

Definition 3.1. (*locally consistent adversaries*) Let $\Pi = (P_1, \ldots, P_n)$ be an n-party protocol and let $\{\alpha^j_{i,i'}\}_{i,i' \in [n], j \in \mathbb{N}}$ be its set of next-message functions, i.e.,

$$m^j_{i,i'} = \alpha^j_{i,i'} \left(b; r; (m^1_{1,i}, \ldots, m^1_{n,i}), \ldots, (m^{j-1}_{1,i}, \ldots, m^{j-1}_{n,i}) \right)$$

is the message party P_i sends to party $P_{i'}$ in the j'th round, given that its input bit is b, the random coins it flipped till now are r, and in round $j' < j$, it got the message $m^{j'}_{i'',i}$ from party $P_{i''}$. An adversary taking the role of P_i is said to be locally consistent with respect to Π, if it *flips its random coins honestly*, and the message it sends in the j'th round to party $P_{i'}$ takes one of the following two forms:

Abort: the message $\perp$.
Input and message selection: a set of messages $\{m_\ell\}^k_{\ell=1}$, for some k, such that for each $\ell \in [k]$:

$$m_\ell = \alpha^j_{i,i'} \left(b_\ell; r; ((m^1_1)_\ell, \ldots, (m^1_n)_\ell), \ldots, ((m^{j-1}_1)_\ell, \ldots, (m^{j-1}_n)_\ell) \right),$$

where $b_\ell \in \{0, 1\}$, r are the coins P_i tossed (honestly) until now, and $(m^{j'}_{i''})_\ell$, for each $j' < j$ and $i'' \neq i$, is one of the messages P_i received from party $P_{i''}$ in the j'th round (or the empty string).

That is, a locally consistent party P_i might send party $P_{i'}$ a sequence of messages (and not just one as instructed), each consistent with a possible choice of its input bit, and some of the messages it received in the previous round. In turn, this will enable party $P_{i'}$, if corrupted, the freedom to choose in the next rounds the message of P_i it would like to act according to. Note that without loss of generality, P_i will always send a single message to the honest parties, as otherwise they will discard the messages.

A few remarks are in place.

1. While the above definition does not enforce between-rounds consistency (a party might send to another party a first-round message consistent with input 0 and a second-round message consistent with input 1), compiling a given protocol so that every message party P_i sends to $P_{i'}$ contains the previous messages P_i sent to $P_{i'}$, will enforce such between-rounds consistency on locally consistent parties.

2. Although a locally consistent adversary tosses its random coins honestly, he may toss all random coins at the beginning of the protocol and choose its actions as a function of these coins. Our attacks in Sects. 4 and 5 do not take advantage of this capability, and let the corrupted parties toss the random coins for a given round at the beginning of the round.

3. Using standard cryptographic techniques, a protocol secure against locally consistent adversaries can be compiled into one secure against arbitrary malicious adversaries, without hurting the efficiency of the protocol "too much," and in particular preserve the round complexity (see Sect. 1.3).

4. The locally consistent parties considered in Sects. 4 and 5 do not take full advantage of the generality of Definition 3.1. Rather, the parties considered either act honestly but abort at the conclusion of the first round, cheat in the first round and then abort, or cheat only in the second round and then abort.

3.1.3. *Public-Randomness Protocols*

In Sect. 1.1, we showed that the description of many natural protocols can be simplified when security is required to hold only against locally consistent adversaries. In this relaxed description a trusted setup phase and cryptographic assumptions are not required, and every party can publish the coins it locally tossed in each round.

Definition 3.2. (*Public-randomness protocols*) A protocol has public randomness, if every party's message consists of two parts: the randomness it sampled in that round, and an arbitrary message which is a function of its view (input, incoming messages, and coins tossed up to and including that point). The party's first message also contains its setup parameters, if such exist.

3.1.4. *Byzantine Agreement*

We now formally define the notion of Byzantine agreement. Since we focus on lower bounds we will consider only the case of a single input bit and a single output bit. A more general notion of Byzantine agreement will include string input and string outputs. A generic reduction shows that the cost of agreeing on strings rather than bits is two additional rounds [68].

Definition 3.3. (*Byzantine Agreement*) We associate the following properties with a PPT n-party Boolean input/output protocol Π.

Agreement. Protocol Π has (t, α)-agreement, if the following holds with respect to any PPT adversary controlling at most t parties in Π and any value of the non-corrupted parties' input bits: in a random execution of Π on sufficiently large security parameter, all non-corrupted parties output the *same* bit with probability at least $1 - \alpha$.[10]

Validity. Protocol Π has (t, β)-validity, if the following holds with respect to any PPT adversary controlling at most t parties in Π and an input bit b given as input to

---

[10]A more general definition would allow the parameter α (and the parameters β, γ below) to depend on the protocol's security parameter. But in this paper we focus on the case that α is a fixed value.

all non-corrupted parties: in a random execution of Π on sufficiently large security parameter, all non-corrupted parties output b with probability at least $1 - \beta$.

Halting. Protocol Π has (t, q, γ)-halting, if the following holds with respect to any PPT adversary controlling at most t parties in Π and any value of the non-corrupted parties' input bits: in a random execution of Π on sufficiently large security parameter, all non-corrupted parties halt within q rounds with probability at least γ.

Protocol Π is a $(t, \alpha, \beta, q, \gamma)$-BA, if it has (t, α)-agreement, (t, β)-validity, and (t, q, γ)-halting. If the protocol has a setup phase, then the above probabilities are taken with respect to this phase as well.

Remark 3.4. (Concrete security) Since we care about fixed values of a protocol's characteristics (i.e., agreement), the role of the security parameter in the above definition is to enable us to bound the running time of the parties and adversaries in consideration in a meaningful way, and to parametrize the cryptographic tools used by the parties (if there are any). Since the attacks we present are efficient assuming the protocol is efficient (in any reasonable sense), the bounds we present are applicable for a fixed protocol that might use a *fixed* cryptographic primitive, e.g., SHA-256.

3.2. *The Bounds*

We proceed to present the formal statements of the three lower bounds. Recall that Byzantine agreement cannot be achieved for $t \geq n/2$, since otherwise the corrupted parties can simply play honestly on an input of their choice and force the output. We therefore consider $t < n/2$ throughout the paper.

First-round halting, arbitrary protocols. The first result bounds the halting probability of arbitrary protocols after a single round. Namely, for "small" values of α and β, the halting probability is "small" for $t \geq n/3$ and "close to 1/2" for $t \geq n/4$.

Theorem 3.5. (restating Theorem 1.2) *Let Π be a PPT n-party protocol that is $(t, \alpha, \beta, 1, \gamma)$-BA against locally consistent, static, non-rushing adversaries. Then,*

- $t \geq n/3$ *implies* $\gamma \leq 6\alpha + 2\beta + \mathsf{err}$
- $t \geq n/4$ *implies* $\gamma \leq 1/2 + 5\alpha + \beta + \mathsf{err}$,

for $\mathsf{err} = 2^{t-n}$ *($\mathsf{err} = 0$ for public-randomness protocols whose security holds against rushing adversaries).*

Second-round halting, arbitrary protocols. The second result bounds the halting probability of arbitrary protocols after two rounds.

Theorem 3.6. (restating Theorem 1.3) *Let Π be a PPT n-party protocol that is $(t, \alpha, \beta, 2, \gamma)$-BA against locally consistent, static, non-rushing adversaries for $t > n/4$. Then $\gamma \leq 1 + 2\alpha + \frac{\beta}{w^2} - \frac{1}{2w^2}$ for $w = \lceil (n - \lceil n/4 \rceil)/ \lfloor t - n/4 \rfloor \rceil + 1$.*

In particular, for $t = (1/4 + \varepsilon) \cdot n$ and "small" α and β, the protocol might not halt at the conclusion of the second round with probability $\approx \varepsilon^2$.

Second-round halting, public-randomness protocols. The third result bounds the halting probability of public-randomness protocols after two rounds. The result requires adaptive and rushing adversaries, and is based on Conjecture 3.8 (stated in Sect. 3.3 below).

Theorem 3.7. (restating Theorem 1.4) *Assume Conjecture 3.8 holds, then for any (constants) $\varepsilon_t, \varepsilon_\gamma > 0$ there exists $\alpha > 0$ such that the following holds for large enough n: let Π be a PPT n-party, public-randomness protocol that is $(t, \alpha, \beta = \varepsilon_\gamma^2/200, 2, \gamma)$-BA against locally consistent, rushing, adaptive adversaries. Then,*

- $t \geq (1/3 + \varepsilon_t) \cdot n$ *implies* $\gamma < \varepsilon_\gamma$.
- $t \geq (1/4 + \varepsilon_t) \cdot n$ *implies* $\gamma < \frac{1}{2} + \varepsilon_\gamma$.

In particular, assuming the protocol has perfect agreement and validity, the protocol never halts in two rounds if the fraction of corrupted parties is greater than $1/3$, and halts in two rounds with probability at most $1/2$ if the fraction of corrupted parties is greater than $1/4$.

The value of α in the theorem is (roughly) $\delta \cdot \varepsilon_t \cdot \varepsilon_\gamma^2$ where δ is the constant guaranteed by Conjecture 3.8. We were not trying to optimize over the constants in the above statement, and in particular it seems that β can be pushed to ε_γ^2.

3.3. *The Combinatorial Conjecture*

Next, we provide the formal statement for the combinatorial conjecture used in Theorem 3.7. For $n \in \mathbb{N}$ and $\sigma \in [0, 1]$, let $\mathbf{D}_{n,\sigma}$ be the distribution induced on the subsets of $[n]$ by sampling each element independently with probability σ. For a finite alphabet Σ, a vector $\boldsymbol{x} \in \Sigma^n$, and a subset $\mathcal{S} \subseteq [n]$, define the vector $\perp_{\mathcal{S}}(\boldsymbol{x}) \in \Sigma^n$ by

$$\perp_{\mathcal{S}}(\boldsymbol{x})_i = \begin{cases} \perp, & i \in \mathcal{S}, \\ \boldsymbol{x}_i, & \text{otherwise.} \end{cases}$$

Conjectures 3.8. (restating Conjecture 1.5) *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,

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

4. Lower Bounds on First-Round Halting

In this section, we present our lower bound for the probability of first-round halting in Byzantine agreement protocols.

Theorem 4.1. (Bound on first-round halting. Theorem 3.5 restated) *Let Π be a* PPT *n-party protocol that is $(t, \alpha, \beta, 1, \gamma)$-*BA *against locally consistent, static, non-rushing adversaries. Then,*

- *$t \geq n/3$ implies $\gamma \leq 6\alpha + 2\beta + $ err*
- *$t \geq n/4$ implies $\gamma \leq 1/2 + 5\alpha + \beta + $ err,*

for err $= 2^{t-n}$ *(*err $= 0$ *for public-randomness protocols whose security holds against rushing adversaries).*

Let Π be as in Theorem 4.1. Without loss of generality and for ease of notation, we denote by Π the modified protocol that outputs $\perp$ if a party does not halt after the first round (it will be clear that the attack, described below, does not benefit from this change). We also omit the security parameter from the parties' input list, it will be clear though that the adversaries we present are efficient with respect to the security parameter.

Lemma 4.2. (Neighboring executions) *Let $v, v' \in \{0, 1\}^n$ be with $\text{dist}(v, v') \leq t$. Then for both $b \in \{0, 1\}$:*

$$\Pr\left[\Pi(v') \in \{b, \perp\}^n \setminus \{\perp^n\}\right] \geq \Pr\left[\Pi(v) \in \{b, \perp\}^n\right] - (1 - \gamma) - 4\alpha - \text{err}.$$

Namely, the lemma bounds from below the probability that in a random honest execution of the protocol on input v', at least one party halts in the first round while outputting b.

We prove Lemma 4.2 below, but first use it to prove Theorem 4.1. We also make use of the following immediate observation.

Claim 4.3. (Almost pre-agreement) *Let $v \in \{0, 1\}^n$ and $b \in \{0, 1\}$ be such that $\text{dist}(v, b^n) \leq t$. Then, $\Pr\left[\Pi(v) \in \{b, \perp\}^n\right] \geq 1 - \alpha - \beta$.*

Proof. Let $\mathcal{A} \subset [n]$ be a subset of size $n - t$ such that $v_{\mathcal{A}} = b^{|\mathcal{A}|}$. The claimed validity of Π yields that

$$\Pr\left[\Pi(v)_{\mathcal{A}} \notin \{b, \perp\}^{|\mathcal{A}|}\right] \leq \beta.$$

This follows from β-validity of Π and the fact that an honest party cannot distinguish between an execution of $\Pi(v)$ and an execution of $\Pi(b^n)$ in which all parties not in $\mathcal{A}$ act as if their input bit is as in v. Hence, by the claimed agreement of Π,

$$\Pr\left[\Pi(v) \notin \{b, \perp\}^n\right] \leq \alpha + \beta.$$

$\square$

Proof of Theorem 4.1.   We separately prove the theorem for $t \geq n/3$ and for $t \geq n/4$.
The case $t \geq n/3$. Let $\boldsymbol{v}_0 = 0^t 1^{\lceil (n-t)/2 \rceil} 0^{\lfloor (n-t)/2 \rfloor}$ and $\boldsymbol{v}_1 = 1^t 1^{\lceil (n-t)/2 \rceil} 0^{\lfloor (n-t)/2 \rfloor}$. Note
that $\mathrm{dist}(\boldsymbol{v}_0, \boldsymbol{v}_1) = t$, and that for both $b \in \{0, 1\}$ it holds that $\mathrm{dist}(\boldsymbol{v}_b, b^n) \leq t$. Hence,
by Claim 4.3, for both $b \in \{0, 1\}$:

$$\Pr\left[\Pi(\boldsymbol{v}_b) \in \{b, \bot\}^n\right] \geq 1 - \alpha - \beta.$$

Applying Lemma 4.2 to $\boldsymbol{v} = \boldsymbol{v}_0$ and $\boldsymbol{v}' = \boldsymbol{v}_1$ yields that

$$\Pr\left[\Pi(\boldsymbol{v}_1) \in \{0, \bot\}^n \setminus \{\bot^n\}\right] \geq \Pr\left[\Pi(\boldsymbol{v}_0) \in \{0, \bot\}^n\right] - (1 - \gamma) - 4\alpha - \mathsf{err}$$

$$\geq 1 - 5\alpha - \beta - (1 - \gamma) - \mathsf{err}.$$

Since by Claim 4.3 it holds that $\Pr\left[\Pi(\boldsymbol{v}_1) \in \{0, \bot\}^n \setminus \{\bot^n\}\right] \leq \Pr\left[\Pi(\boldsymbol{v}_1) \notin \{1, \bot\}^n\right] \leq$
$\alpha + \beta$, we conclude that $6\alpha + 2\beta + (1 - \gamma) + \mathsf{err} \geq 1$, hence $\gamma \leq 6\alpha + 2\beta + \mathsf{err}$.
The case $t \geq n/4$. In this case there are no two vectors that are t apart in Hamming
distance, and still each of them has $n-t$ entries of opposite values. Rather, we consider the
two vectors $\boldsymbol{v}_0 = 0^t 0^t 0^t 1^{n-3t}$ and $\boldsymbol{v}_1 = 1^t 1^t 0^t 1^{n-3t}$ of distance $2t$. For both $b \in \{0, 1\}$,
the vector $\boldsymbol{v}_b$ has at least $n - t$ entries with b and is of distance t from the vector
$\mathbf{v}^\star = 1^t 0^t 0^t 1^{n-3t}$.
   As in the first part of the proof, Applying Claim 4.3 and Lemma 4.2 on $\boldsymbol{v}_b$ and $\mathbf{v}^\star$, for
both $b \in \{0, 1\}$, yields that

$$\Pr\left[\Pi(\mathbf{v}^\star) \in \{b, \bot\}^n \setminus \{\bot^n\}\right] \geq \Pr\left[\Pi(\boldsymbol{v}_b) \in \{b, \bot\}^n\right] - (1 - \gamma) - 4\alpha - \mathsf{err}$$

$$\geq 1 - 5\alpha - \beta - (1 - \gamma) - \mathsf{err}.$$

By union bound, we conclude that $2(5\alpha + \beta + (1 - \gamma) + \mathsf{err}) \geq 1$, hence $\gamma \leq 1/2 + 5\alpha + \beta + \mathsf{err}$. □

4.1. *Proving Lemma 4.2*

Proof of Lemma 4.2.   Fix $b \in \{0, 1\}$ and let $\delta = \Pr\left[\Pi(\boldsymbol{v}) \in \{b, \bot\}^n\right]$. Let $\mathcal{P}$ be the
coordinates in which $\boldsymbol{v}$ and $\boldsymbol{v}'$ differ, and let $\overline{\mathcal{P}} = [n] \setminus \mathcal{P}$. Let I be the index (a function
of the parties' coins and setup parameters) of the smallest party in $\overline{\mathcal{P}}$ that halts in the first
round and outputs *the same value*, both if the parties in $\mathcal{P}$ send their messages according
to input $\boldsymbol{v}$ and if they do that according to $\boldsymbol{v}'$. We let $I = 0$ if there is no such party, and
(abusing notation) sometimes identify I with the event that $I \neq 0$, e.g., $\Pr[I]$ stands for
$\Pr[I \neq 0]$. By definition,

$$\delta \leq \Pr\left[\Pi(\boldsymbol{v}) \in \{b, \bot\}^n \quad \wedge \quad I\right] + (1 - \Pr[I])$$

and thus

$$\Pr\left[\Pi(\boldsymbol{v}) \in \{b, \bot\}^n \quad \wedge \quad I\right] \geq \delta - (1 - \Pr[I]) \tag{8}$$

It follows that

$$
\begin{aligned}
\Pr\left[\Pi(\boldsymbol{v}') \in \{b, \bot\}^n \setminus \{\bot^n\}\right] &\geq \Pr\left[\Pi(\boldsymbol{v}') \in \{b, \bot\}^n \ \wedge \ I\right] \\
&= \Pr\left[\Pi(\boldsymbol{v}') \in \{b, \bot\}^n \ \wedge \ \Pi(\boldsymbol{v}')_I = b\right] \\
&\geq \Pr\left[\Pi(\boldsymbol{v}')_I = b\right] - \alpha \\
&= \Pr\left[\Pi(\boldsymbol{v})_I = b\right] - \alpha \\
&\geq \Pr\left[\Pi(\boldsymbol{v}) \in \{b, \bot\}^n \ \wedge \ \Pi(\boldsymbol{v})_I = b\right] - 2\alpha \\
&= \Pr\left[\Pi(\boldsymbol{v}) \in \{b, \bot\}^n \ \wedge \ I\right] - 2\alpha \\
&\geq \delta - (1 - \Pr[I]) - 2\alpha. \tag{9}
\end{aligned}
$$

The first inequality and the equalities hold by the definition of I. The second and third inequalities hold by agreement, and the last inequality holds by Eq. (8). We conclude the proof showing that:

$$
\Pr[I] \geq \gamma - \mathsf{err} - 2\alpha \tag{10}
$$

Let C denote the event (a function of the parties' coins and setup parameters) that for each party j in $\overline{\mathcal{P}}$ there exists an input in $\{\boldsymbol{v}, \boldsymbol{v}'\}$ on which it does not halt. Furthermore, let $F := \neg C \wedge (I = 0)$, i.e., there exists a party that halts on both inputs but outputs different values. By definition, $I = 0$ is equivalent to the event $F \vee C$.

Consider the adversary that in the first round acts toward a random subset $\overline{\mathcal{P}}' \subseteq \overline{\mathcal{P}}$ according to input $\boldsymbol{v}$, towards the remaining parties according to $\boldsymbol{v}'$, and aborts at the end of this round. Fix some random coins and setup parameters in F, and let $i \in \overline{\mathcal{P}}$ be a party that, under this fixing, halts in the first round on both $\boldsymbol{v}$ and $\boldsymbol{v}'$, but outputs a different value. Note that the other parties in $\overline{\mathcal{P}}$ cannot distinguish whether i is in $\overline{\mathcal{P}}'$ or not (in both cases i halts at the end of the first round). Since, by assumption, $t < n/2$ (i.e., there exist additional honest parties), it follows that under the above conditioning, agreement is violated with probability at least $1/2$. We conclude that $\Pr[F] \leq 2\alpha$.

It is also clear that when C occurs, the above attacker fails to prevent an honest party in $\overline{\mathcal{P}}$ from halting in the first round only if the following event happens: each party in $\overline{\mathcal{P}}$ does not halt in $\Pi(\boldsymbol{v}'')$ for some $\boldsymbol{v}'' \in \{\boldsymbol{v}, \boldsymbol{v}'\}$, but the adversary acts towards each of these parties on the input in which it does halt. The latter event happens with probability at most $2^{-|\overline{\mathcal{P}}|} \leq 2^{t-n} = \mathsf{err}$. Thus, $\Pr[C] \leq 1 - (\gamma - \mathsf{err})$. We conclude that

$$
\Pr[I] \geq 1 - \Pr[C] - \Pr[F] \geq \gamma - \mathsf{err} - 2\alpha \tag{11}
$$

Finally, we note that if the protocol has public randomness, the (now rushing) attacker does not have to guess what input to act upon. Rather, after seeing the first-round randomness, it *finds* an input $\boldsymbol{v}'' \in \{\boldsymbol{v}, \boldsymbol{v}'\}$ such that at least one party in $\overline{\mathcal{P}}$ does not halt in $\Pi(\boldsymbol{v}'')$ or violates agreement, and acts according to this input. Specifically, given

the honest parties' first-round coins, the attacker can compute on its own all honest-to-honest first-round messages (recall that we consider private channels, so the attacker does not see those messages on the channels), and locally check which honest party will halt with output 0 and which will halt with output 1 when playing according to v and when playing according to v'. Hence, the bound on I changes to

$$\Pr[I] \geq \gamma - \alpha,$$

proving the theorem statement for such protocols.   □

5. Lower Bounds on Second-Round Halting

In this section, we prove lower bounds for second-round halting of Byzantine agreement protocols. In Sect. 5.1, we prove a bound for arbitrary protocols, and in Sect. 5.2, we give a much stronger bound for public-randomness protocols (the natural extension of public-coin protocols to the "with-input" setting).

5.1. Arbitrary Protocols

We start by proving our lower bound for second-round halting of arbitrary protocols.

Theorem 5.1. (Bound on second-round halting, arbitrary protocols. Theorem 3.6 restated) *Let Π be a* PPT *n-party protocol that is $(t, \alpha, \beta, 2, \gamma)$-*BA *against locally consistent, static, non-rushing adversaries for $t > n/4$. Then $\gamma \leq 1 + 2\alpha + \frac{\beta}{w^2} - \frac{1}{2w^2}$ for $w = \lceil (n - \lceil n/4 \rceil)/ \lfloor t - n/4 \rfloor \rceil + 1$.*

Let Π be as in Theorem 5.1. Without loss of generality and for ease of notation, we denote by Π the modified protocol that outputs $\perp$ if a party does not halt after the first two rounds (it will be clear that the attack, described below, does not benefit from this change). We also assume without loss of generality that the honest parties in an execution of Π never halt in the first round (by adding a dummy round if needed). Finally, we omit the security parameter from the parties' input list, it will be clear though that the adversaries we present are efficient with respect to the security parameter.

Let $k = \lceil n/4 \rceil$ and let $h = \lceil (n - k)/(t - k) \rceil$. The theorem is easily implied by the next lemma.

Lemma 5.2. (Neighboring executions) *Let $v, v' \in \{0, 1\}^n$ be with $\text{dist}(v, v') \leq k$. Then, for every $b \in \{0, 1\}$:*

$$\Pr\left[\Pi(v') = b^n\right] \geq \Pr\left[\Pi(v) = b^n\right] - h(h + 1)(2\alpha + 1 - \gamma) - \alpha.$$

Namely, the lemma bounds from below the probability that in a random honest execution of the protocol on input v' all parties halt within two rounds while outputting b.

We prove Lemma 5.2 below, but first use it to prove Theorem 5.1. We also make use of the following immediate observation.

Claim 5.3. (Almost pre-agreement) *Let $v \in \{0, 1\}^n$ and $b \in \{0, 1\}$ be such that* $\text{dist}(v, b^n) \leq t$. *Then,* $\Pr\left[\Pi(v) = b^n\right] \geq 1 - \alpha - \beta - (1 - \gamma)$.

Proof. The same argument as in the proof of Claim 4.3 yields that

$$\Pr\left[\Pi(v) \notin \{b, \bot\}^n\right] \leq \alpha + \beta.$$

Thus, by γ-second-round halting

$$\Pr\left[\Pi(v) \neq b^n\right] \leq \alpha + \beta + (1 - \gamma).$$

<div align="right">□</div>

Proof of Theorem 5.1. Consider the vectors $v_0 = 0^k 0^k 0^k 1^{n-3k}$, $v_1 = 1^k 1^k 0^k 1^{n-3k}$ and $v^\star = 1^k 0^k 0^k 1^{n-3k}$. Note that for both $b \in \{0, 1\}$ it holds that $\text{dist}(v_b, b^n) \leq t$ since $n/4 \leq k \leq t$), and that $\text{dist}(v_b, v^\star) = k$. Applying Lemma 5.2 and Claim 5.3 for each of these vectors, yields that for both $b \in \{0, 1\}$:

$$\Pr\left[\Pi(v^\star) = b^n\right] \geq 1 - \alpha - \beta - (1 - \gamma) - h(h + 1)(2\alpha + 1 - \gamma) - \alpha$$

$$\geq 1 - \beta - (h + 1)^2(2\alpha + 1 - \gamma).$$

Note that $w = h + 1$, which implies $\beta + w^2(2\alpha + 1 - \gamma) \geq 1/2$, and the proof follows by a simple calculation. □

5.1.1. *Proving Lemma 5.2*

We assume for ease of notation that $\text{dist}(v, v') = k$ (rather than $\leq k$) and let $\ell = t - k$. Assume for ease of notation that $h \cdot \ell = n - k$ (i.e., no rounding), and for a k-size subset of parties $\mathcal{P} \subset [n]$, let $\mathcal{L}_1^{\mathcal{P}}, \ldots, \mathcal{L}_h^{\mathcal{P}}$ be an arbitrary partition of $\overline{\mathcal{P}} = [n] \setminus \mathcal{P}$ into ℓ-size subsets. Consider the following family of protocols:

Protocol 5.4. $(\Pi_d^{\mathcal{P},\mathcal{S}})$

Parameters: *A subset $\mathcal{P} \subseteq [n]$ and an index $d \in (h)$.*
Input: *Every party P_i has an input bit $v_i \in \{0, 1\}$.*
First round:

> **Party $\mathsf{P}_i \in \mathcal{P}$.** *If $d = 0$ [resp., $d = h$], act honestly according to Π with respect to input bit v_i [resp., $1 - v_i$]. Otherwise,*
>
>> 1. *Choose random coins honestly (i.e., uniformly at random).*
>> 2. *To each party in $\bigcup_{j \in \{1,\ldots,d\}} \mathcal{L}_j^{\mathcal{P}}$: send a message according to input $1 - v_i$.*
>> 3. *To each party in $\bigcup_{j \in \{d+1,\ldots,h\}} \mathcal{L}_j^{\mathcal{P}}$: send a message according to input v_i (real input).*
>> 4. *Send no messages to the other parties in $\mathcal{P}$.*
>
> **Other parties.** *Act according to Π.*

Second round:

> **Party $\mathsf{P}_i \in \mathcal{P}$.** *If $d = 0$ [resp., $d = h$], act honestly according to Π with respect to input bit v_i [resp., $1 - v_i$]; otherwise, abort.*
> **Other parties.** *Act honestly according to Π.*

. . . .

Namely, the "pivot" parties in $\mathcal{P}$ gradually shift their inputs from their real input to its negation according to parameter d. Note that protocol $\Pi_0^{\mathcal{P},\mathcal{S}}(\boldsymbol{v})$ is equivalent to an honest execution of protocol $\Pi(\boldsymbol{v})$, and $\Pi_h^{\mathcal{P},\mathcal{S}}(\boldsymbol{v})$ is equivalent to an honest execution of $\Pi(\boldsymbol{v}')$, for $\boldsymbol{v}'$ being $\boldsymbol{v}$ with the coordinates in $\mathcal{P}$ negated. Note that for "intermediately" protocols $\Pi_d^{\mathcal{P},\mathcal{S}}$ for $0 < d < h$, the pivot parties send conflicting messages to honest parties in the first round and abort in the second round. The reason that aborting in the second round does not affect our analysis below is that, without loss of generality, honest parties can exchange their views in the second round, realize the pivot parties are cheating (as we consider locally consistent adversaries), and ignore their messages. Lemma 5.2 easily follows by the next claim about Protocol 5.4. In the following we let $\delta_b = \Pr\left[\Pi(\boldsymbol{v})_{\overline{\mathcal{P}}} = b^{|\overline{\mathcal{P}}|}\right]$.

Claim 5.5. *For every k-size subset $\mathcal{P} \subset [n]$, $b \in \{0, 1\}$ and $d \in (h)$, it holds that*

$$\Pr\left[\Pi_d^{\mathcal{P},\mathcal{S}}(\boldsymbol{v})_{\overline{\mathcal{P}}} = b^{|\overline{\mathcal{P}}|}\right] \geq \delta_b - d(h+1)(2\alpha + 1 - \gamma).$$

We prove Claim 5.5 below, but first use it to prove Lemma 5.2.

Proof of Lemma 5.2. By Claim 5.5,

$$\Pr\left[\Pi_h^{\mathcal{P},\mathcal{S}}(\boldsymbol{v})_{\overline{\mathcal{P}}} = b^{|\overline{\mathcal{P}}|}\right] \geq \delta_b - h(h+1)(2\alpha + 1 - \gamma).$$

Recall that $\Pi_h^{\mathcal{P},\mathcal{S}}(v)_{\overline{\mathcal{P}}} = b^{\left|\overline{\mathcal{P}}\right|}$ only when parties complete the protocol in the second round, since, by assumption, a party that continues to beyond the second round outputs $\perp$. In addition, since $\Pi_h^{\mathcal{P},\mathcal{S}}(v)$ is just an honest execution of $\Pi(v')$, by agreement it holds that

$$\Pr\left[\Pi(v') = b^n\right] \geq \delta_b - h(h+1)(2\alpha + 1 - \gamma) - \alpha.$$

$\square$

Proof of Claim 5.5.   The proof is by induction on d. The base case $d = 0$ holds by definition. Suppose for contradiction the claim does not hold, and let $d^* \in (h-1)$ be such that the claim holds for d^* but not for $d^* + 1$. Let γ_d be the probability that all honest parties halt in the second round of a random execution of $\Pi_d^{\mathcal{P},\mathcal{S}}(v)$. Since the claim holds for d^*, it holds that

$$\Pr\left[\Pi_{d^*}^{\mathcal{P},\mathcal{S}}(v)_{\overline{\mathcal{P}}} = b^{\left|\overline{\mathcal{P}}\right|}\right] \geq \delta_b - \beta - d^*(h+1)(2\alpha + 1 - \gamma) \tag{12}$$

Since the claim does not hold for $d^* + 1$, but all honest parties output something in $\Pi_{d^*+1}^{\mathcal{P},\mathcal{S}}$ with probability at least γ_{d^*+1}, we have that

$$\Pr\left[\Pi_{d^*+1}^{\mathcal{P},\mathcal{S}}(v)_{\overline{\mathcal{P}}} \in \{0,1\}^{\left|\overline{\mathcal{P}}\right|} \setminus \{b^{\left|\overline{\mathcal{P}}\right|}\}\right] > 1$$
$$- \left(\delta_b - \beta - (d^*+1)(h+1)(2\alpha + 1 - \gamma)\right) - (1 - \gamma_{d^*+1}) \tag{13}$$

We note that for every $d \in (h)$

$$\frac{1 - \gamma_d}{h+1} \leq 1 - \gamma \tag{14}$$

Indeed, otherwise, the adversary that corrupts the parties in $\mathcal{P}$ and acts like $\Pi_d^{\mathcal{P},\mathcal{S}}$ for a random $d \in (h)$, violates the γ-second-round-halting property of Π. We conclude that

$$\Pr_r\left[\Pi_{d^*}^{\mathcal{P},\mathcal{S}}(v;r)_{\overline{\mathcal{P}}} = b^{\left|\overline{\mathcal{P}}\right|} \quad \wedge \quad \Pi_{d^*+1}^{\mathcal{P},\mathcal{S}}(v;r)_{\overline{\mathcal{P}}} \in \{0,1\}^{\left|\overline{\mathcal{P}}\right|} \setminus \{b^{\left|\overline{\mathcal{P}}\right|}\}\right]$$
$$\geq 1 - \left(1 - \Pr_r\left[\Pi_{d^*}^{\mathcal{P},\mathcal{S}}(v;r)_{\overline{\mathcal{P}}} = b^{\left|\overline{\mathcal{P}}\right|}\right]\right)$$
$$- \left(1 - \Pr_r\left[\Pi_{d^*+1}^{\mathcal{P},\mathcal{S}}(v;r)_{\overline{\mathcal{P}}} \in (\{0,1\}^{\left|\overline{\mathcal{P}}\right|} \setminus \{b^{\left|\overline{\mathcal{P}}\right|}\})\right]\right)$$
$$> (h+1)(2\alpha + 1 - \gamma) - (1 - \gamma_{d^*+1})$$
$$\geq 2\alpha(h+1), \tag{15}$$

for r being the randomness of the parties. The second inequality is by Eq. (12) and (13), and the third one by Eq. (14).

Consider the adversary A that samples $d \leftarrow (h-1)$, corrupts the parties in $\mathcal{P} \cup \mathcal{L}_{d+1}^{\mathcal{P}}$, and acts towards a uniform random subset of the honest parties according to $\Pi_d^{\mathcal{P},\mathcal{S}}$ and to the remaining parties according to $\Pi_{d+1}^{\mathcal{P},\mathcal{S}}$. Since A violates agreement if it guesses $d = d^*$ and it partitions the honest parties suitably, Eq. (15) yields that A causes disagreement with probability larger than $2\alpha(h+1)/(2(h+1)) = \alpha$. Since A corrupts $|\mathcal{P} \cup \mathcal{L}_{d+1}^{\mathcal{P}}| \leq t$ parties, this contradicts the assumption about Π. □

5.2. *Public-Randomness Protocols*

We proceed to prove our lower bound for second-round halting of public-randomness protocols.

Theorem 5.6. (Lower bound on second-round halting, public-randomness protocols. Theorem 3.7 restated) *Assume Conjecture 3.8 holds, then for any (constants) $\varepsilon_t, \varepsilon_\gamma > 0$ there exists $\alpha > 0$ such that the following holds for large enough n: let Π be a PPT n-party, public-randomness protocol that is $(t, \alpha, \beta = \varepsilon_\gamma^2/200, 2, \gamma)$-BA against locally consistent, rushing, adaptive adversaries. Then,*

- *$t \geq (1/3 + \varepsilon_t) \cdot n$ implies $\gamma < \varepsilon_\gamma$.*
- *$t \geq (1/4 + \varepsilon_t) \cdot n$ implies $\gamma < \frac{1}{2} + \varepsilon_\gamma$.*

Assume Conjecture 3.8 holds. Let Π be as in the theorem statement, and assume $\gamma = \varepsilon_\gamma$ in the case $t \geq (1/3 + \varepsilon_t) \cdot n$ and $\gamma = \frac{1}{2} + \varepsilon_\gamma$ in the case $t \geq (1/4 + \varepsilon_t) \cdot n$. Let $\lambda = \varepsilon_\gamma/10$ and $\sigma = \varepsilon_t/4$. Recall that $\perp_{\mathcal{S}}(x)$ is the string resulting by replacing all entries of x indexed by $\mathcal{S}$ with $\perp$. Conjecture 3.8 yields that there exists $\delta > 0$ such that the following holds for large enough n: let Σ be a finite alphabet and let $\mathcal{A}_0, \mathcal{A}_1 \subset \{\Sigma \cup \perp\}^n$ be two sets such that for both $b \in \{0, 1\}$:

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

Then,

$$\Pr_{r \leftarrow \Sigma^n, \mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} [\forall b \in \{0, 1\}: \{r, \perp_{\mathcal{S}}(r)\} \cap \mathcal{A}_b \neq \emptyset] \geq \delta. \tag{16}$$

In the following we assume $\alpha = \min \{\delta\lambda\varepsilon_t/10, \beta\}$ and derive a contradiction, yielding that the agreement error has to be larger than that.

Fix n that is large enough for Eq. (16) to hold and that (by Chernoff bound) $\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} [|\mathcal{S}| > 2\sigma n] = 2^{-\Theta(n \cdot \sigma)} \leq \alpha$, i.e., $n > \Theta((\log 1/\alpha)/\sigma)$. As in the proof of Theorem 5.1, we assume for ease of notation that an honest party that runs more than two round outputs $\perp$, and that the honest parties in Π never halt in one round. We also omit the security parameter from the parties input list. We assume without loss of generality that in the first round, the parties flip no coin, since such coins can be added to the setup parameter.

We use the following notation: the setup parameter and second-round randomness of the parties in Π are identified with elements of $\mathcal{F}$ and $\mathcal{R}$, respectively. We denote

by f_i and r_i the setup parameter and the second-round randomness of party P_i in Π, and let $D_{\mathcal{F}}$ be the joint distribution of the parties' setup parameters (by definition, the joint distribution of the second-round randomness is the product distribution $\mathcal{R}^n$). For $v \in \{0, 1\}^n$, $f = (f_1, \ldots, f_n) \in \text{Supp}(D_{\mathcal{F}})$, and $r = (r_1, \ldots, r_n) \in \mathcal{R}^n$, let $\Pi(v; (f, r))$ denote the execution of Π in which party P_i gets input v_i, setup parameter f_i and second-round randomness r_i. We naturally apply this notation for the variants of Π considered in the proof.

For $\mathcal{S} \subseteq [n]$, let $\Pi^{\mathcal{S}}$ be the variant of Π in which the parties in $\mathcal{S}$ halt at the end of the first round. Let $k = \lceil t - \varepsilon_t \cdot n \rceil$ (i.e., $k = \lceil n/3 \rceil$ if $t \geq (1/3 + \varepsilon_t) \cdot n$, and $k = \lceil n/4 \rceil$ if $t \geq (1/4 + \varepsilon_t) \cdot n$). The heart of the proof lies in the following lemma.

Lemma 5.7. (Neighboring executions) *Let $v, v' \in \{0, 1\}^n$ be with $\text{dist}(v, v') \leq k$, let $b \in \{0, 1\}$, and let $\overline{\mathcal{S}} = [n] \setminus \mathcal{S}$. Then, with probability at least $\gamma - 7\lambda - \frac{\alpha + \Pr[\Pi(v) \neq b^n]}{\lambda}$ over $f \leftarrow D_{\mathcal{F}}$, it holds that*

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\Pr_{r \leftarrow \mathcal{R}^n} \left[\Pi(v'; (f, r)) = b^n \ \wedge \ \Pi^{\mathcal{S}}(v'; (f, r))_{\overline{\mathcal{S}}} = b^{\left|\overline{\mathcal{S}}\right|} \right] \geq \lambda \right] \geq 1 - \delta.$$

Namely, in an execution of $\Pi(v')$, all honest parties halt after two rounds and output b, regardless of whether a random subset of parties aborts after the first round. Lemma 5.7 is proven in Sect. 5.2.1, but let us first use it to prove Theorem 5.6. We make use of the following immediate observation:

Claim 5.8. (Almost pre-agreement) *Let $v \in \{0, 1\}^n$ and $b \in \{0, 1\}$ be such that $\text{dist}(v, b^n) \leq t$. Then, $\Pr\left[\Pi(v) \in \{b, \perp\}^n\right] \geq 1 - \alpha - \beta$.*

Proof. The proof of this claim uses an identical argument as in the proof of Claim 4.3. $\square$

Proving Theorem 5.6.

Proof of Theorem 5.6. We separately prove the case $t \geq (1/3 + \varepsilon_t) \cdot n$ and $t \geq (1/4 + \varepsilon_t) \cdot n$.

The case $t \geq (1/3 + \varepsilon_t) \cdot n$. Let $v_0 = 0^k 1^{\lceil (n-k)/2 \rceil} 0^{\lfloor (n-k)/2 \rfloor}$ and let $v_1 = 1^k 1^{\lceil (n-k)/2 \rceil} 0^{\lfloor (n-k)/2 \rfloor}$. Note that $\text{dist}(v_0, v_1) = k$ and that for both $b \in \{0, 1\}$ it holds that $\text{dist}(v_b, b^n) \leq t$. We will use Lemma 5.7 and Claim 5.8 to prove that $\Pi(v_1) = 0^n$ with noticeable probability, contradicting the validity of the protocol.

Recall that, in this case, $\gamma = \varepsilon_\gamma$, that $\lambda = \varepsilon_\gamma/10$ and $\alpha, \beta \leq \varepsilon_\gamma^2/200 = \lambda^2/2$. Claim 5.8 yields that for both $b \in \{0, 1\}$:

$$\Pr\left[\Pi(v_b) \neq \overline{b}^n\right] \geq \Pr\left[\Pi(v_b) \in \{b, \perp\}^n\right] \geq 1 - \alpha - \beta \geq 1 - \lambda^2 \qquad (17)$$

Applying Lemma 5.7 with respect to $\boldsymbol{v}_0$ and $\boldsymbol{v}_1$ and $b = 0$, yields that with probability at least

$$\gamma - 7\lambda - \frac{\alpha + \Pr\left[\Pi(\boldsymbol{v}_0) \neq 0^n\right]}{\lambda} \geq 3\lambda - \lambda = 2\lambda$$

over $\boldsymbol{f} \leftarrow D_{\mathcal{F}}$, it holds that (by discarding the probability over $\mathcal{S}$ since the item below does not depend on $\mathcal{S}$)

$$\Pr_{\boldsymbol{r}}\left[\Pi(\boldsymbol{v}_1; (\boldsymbol{f}, \boldsymbol{r})) = 0^n\right] \geq \lambda.$$

Therefore, overall

$$\Pr\left[\Pi(\boldsymbol{v}_1) = 0^n\right] \geq 2\lambda^2,$$

in contradiction to Eq. (17).

The case $t \geq (1/4 + \varepsilon_t) \cdot n$. Consider the vectors $\boldsymbol{v}_0 = 0^k 0^k 0^k 1^{n-3k}$, $\boldsymbol{v}_1 = 1^k 1^k 0^k 1^{n-3k}$ and $\boldsymbol{v}^\star = 1^k 0^k 0^k 1^{n-3k}$. Note that for both $b \in \{0, 1\}$ it holds that $\text{dist}(\boldsymbol{v}_b, b^n) \leq t$ and that $\text{dist}(\boldsymbol{v}_b, \boldsymbol{v}^\star) = k$. Applying Lemma 5.7 and Claim 5.8 on $\boldsymbol{v}_b$ and $\boldsymbol{v}^\star$, for both $b \in \{0, 1\}$, yields that $\Pi^{\mathcal{S}}(\boldsymbol{v}^\star) = b^n$ with noticeable probability over the choice of $\mathcal{S}$. This will allow us to use Conjecture 3.8 to lowerbound the protocol's agreement.

Recall that the distribution $\mathbf{D}_{n,\sigma}$, from which set $\mathcal{S}$ is sampled, is the distribution induced on the subsets of $[n]$ by sampling each element independently with probability σ. In addition, recall that in the case at hand ($t \geq (1/4 + \varepsilon_t) \cdot n$), we assume that $\gamma = 1/2 + \varepsilon_\gamma$. A similar calculation to the previous case yields that by Lemma 5.7 and Claim 5.8, for both $b \in \{0, 1\}$: with probability at least $\frac{1}{2} + 2\lambda$ over $\boldsymbol{f} \leftarrow D_{\mathcal{F}}$ it holds that

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}}\left[\Pr_{\boldsymbol{r} \leftarrow \mathcal{R}^n}\left[\Pi(\boldsymbol{v}^\star; (\boldsymbol{f}, \boldsymbol{r})) = b^n \ \wedge \ \Pi^{\mathcal{S}}(\boldsymbol{v}^\star; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{S}}} = b^{\left|\overline{\mathcal{S}}\right|}\right] \geq \lambda\right] \geq 1 - \delta.$$

It follows that there exists a set $\mathcal{T} \subseteq \text{Supp}(D_{\mathcal{F}})$ with $\Pr_{\boldsymbol{f} \leftarrow D_{\mathcal{F}}}[\mathcal{T}] \geq 4\lambda$, such that for every $\boldsymbol{f} \in \mathcal{T}$, for *both* $b \in \{0, 1\}$:

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}}\left[\Pr_{\boldsymbol{r} \leftarrow \mathcal{R}^n}\left[\Pi(\boldsymbol{v}^\star; (\boldsymbol{f}, \boldsymbol{r})) = b^n \ \wedge \ \Pi^{\mathcal{S}}(\boldsymbol{v}^\star; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{S}}} = b^{\left|\overline{\mathcal{S}}\right|}\right] \geq \lambda\right] \geq 1 - \delta \tag{18}$$

We assume without loss of generality that if a party gets $\bot$ as its second-round random coins, it aborts after the first round. For $\boldsymbol{r} \in (\mathcal{R} \cup \{\bot\})^n$ let $\mathcal{E}(\boldsymbol{r})$ be the indices in $\boldsymbol{r}$ of the value $\bot$. For $\boldsymbol{f} \in \text{Supp}(D_{\mathcal{F}})$ and $b \in \{0, 1\}$, let

$$\mathcal{A}_b^f = \left\{\boldsymbol{r} \in \{\mathcal{R} \cup \{\bot\}\} : \Pi(\boldsymbol{v}^\star; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{E}(\boldsymbol{r})}} = b^{\left|\overline{\mathcal{E}(\boldsymbol{r})}\right|}\right\} \tag{19}$$

By Eq. (18), for $f \in \mathcal{T}$ and $b \in \{0, 1\}$, it holds that

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\Pr_{r \leftarrow \mathcal{R}^n} \left[r, \perp_{\mathcal{S}}(r) \in \mathcal{A}_b^f \right] \geq \lambda \right] \geq 1 - \delta \qquad (20)$$

Hence by Conjecture 3.8, see Eq. (16), for $f \in \mathcal{T}$ it holds that

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

That is,

$$\Pr_{r \leftarrow \mathcal{R}^n, \mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\forall b \in \{0, 1\} \ \ \exists \mathcal{S}_b \in \{\mathcal{S}, \emptyset\} : \Pi^{\mathcal{S}_b}(\mathbf{v}^\star; (f, r))_{\overline{\mathcal{S}_b}} = b^{\left| \overline{\mathcal{S}_b} \right|} \right] > \delta \quad (21)$$

Consider the following adversary:

Algorithm 5.9. *(A)*

Pre-interaction. *Corrupt a random subset $\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}$ conditioned on $|\mathcal{S}| \leq 2\sigma n$.*
First round. *Act according to Π.*
Second round. *Sample $\mathcal{S}_0, \mathcal{S}_1$ at random from $\{\emptyset, \mathcal{S}\}$, and act towards some honest parties according to $\Pi^{\mathcal{S}_0}$ and towards the others according to $\Pi^{\mathcal{S}_1}$.*

Recall that n is chosen so that $\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} [|\mathcal{S}| > 2\sigma n] \leq \alpha$ and that $\alpha < \delta/2$. By Eq. (21), the above adversary violates the agreement of Π on input $\mathbf{v}^\star$ with probability larger than $\delta - \Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} [|\mathcal{S}| > 2\sigma n] \geq \delta - \alpha > \alpha$, in contradiction with the assumed agreement of Π. $\qquad\qquad\square$

5.2.1. *Proving Lemma 5.7*

Fix $v, v' \in \{0, 1\}^n$ and $b \in \{0, 1\}$ as in the lemma statement. We assume for simplicity that $\text{dist}(v, v') = k$ (rather than $\leq k$). Let $\ell = \lfloor (t - k)/2 \rfloor$ and let $h = \lceil (n - k)/\ell \rceil$. Assume for ease of notation that $h \cdot \ell = n - k$ (i.e., no rounding), and for a k-size subset of parties $\mathcal{P} \subset [n]$, let $\mathcal{L}_1^{\mathcal{P}}, \ldots, \mathcal{L}_h^{\mathcal{P}}$ be an arbitrary partition of $\overline{\mathcal{P}} = [n] \setminus \mathcal{P}$ into ℓ-size subsets. Consider the following protocol family.

Protocol 5.10. $(\Pi_d^{\mathcal{P},\mathcal{S}})$

Parameters: *subsets* $\mathcal{P}, \mathcal{S} \subseteq [n]$ *and an index* $d \in (h)$.
Input: *Party* P_i *has a setup parameter* f_i *and an input bit* v_i.
First round:

> **Party** $\mathsf{P}_i \in \mathcal{P}$. *If* $d = 0$ *[resp.,* $d = h$*], act honestly according to* Π *with respect to input bit* v_i *[resp.,* $1 - v_i$*]. Otherwise,*
>
> > 1. *Choose random coins honestly (i.e., uniformly at random).*
> > 2. *To each party in* $\bigcup_{j \in \{1, \dots, d\}} \mathcal{L}_j^{\mathcal{P}}$: *send a message according to input* $1 - v_i$.
> > 3. *To each party in* $\bigcup_{j \in \{d+1, \dots, h\}} \mathcal{L}_j^{\mathcal{P}}$: *send a message according to input* v_i *(real input).*
> > 4. *Send no messages to the other parties in* $\mathcal{P}$.
>
> **Other parties.** *Act according to* Π.

Second round:

> **Parties in** $\mathcal{P} \setminus \mathcal{S}$. *If* $d = 0$ *[resp.,* $d = h$*], act honestly according to* Π *with respect to input bit* v_i *[resp.,* $1 - v_i$*]; otherwise, abort.*
> **Parties in** $\mathcal{S}$. *Abort.*
> **Other parties.** *Act according to* Π.

Namely, the "pivot" parties in $\mathcal{P}$ shift their inputs from their real input to the flipped one according to parameter d. The "aborting" parties in $\mathcal{S}$ abort at the end of the first round. Note that protocol $\Pi_0^{\mathcal{P},\mathcal{S}}$ is the same as protocol $\Pi^{\mathcal{S}}$, and $\Pi_h^{\mathcal{P},\mathcal{S}}(v)$ acts like $\Pi^{\mathcal{S}}(v')$, for v' being v with the coordinates in $\mathcal{P}$ flipped.

For $\mathcal{P}, \mathcal{S} \subseteq [n]$, let $\overline{\mathcal{P} \cup \mathcal{S}} = [n] \setminus (\mathcal{P} \cup \mathcal{S})$, let $d \in (h)$, let $c \in \{0, 1\}$, and let

$$
\mathcal{V}_{d,c}^{\mathcal{P}} = \left\{ (f, \mathcal{S}, r) : \quad \Pi_d^{\mathcal{P},\mathcal{S}}(v; (f, r))_{\overline{\mathcal{P} \cup \mathcal{S}}} = c^{\left|\overline{\mathcal{P} \cup \mathcal{S}}\right|} \right\}.
$$

Namely, $\mathcal{V}_{d,c}^{\mathcal{P}}$ are the sets, setup parameters and random strings on which honest parties in $\Pi_d^{\mathcal{P},\mathcal{S}}$ halt in the second round and output c. Let $\chi = \Pr\left[\Pi(v) \neq b^n\right]$ and let

$$
\mathcal{T}_{d,c}^{\mathcal{P}} = \left\{ f : \Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\Pr_{r \leftarrow \mathcal{R}^n} \left[(f, \mathcal{S}, r), (f, \emptyset, r) \in \mathcal{V}_{d,c}^{\mathcal{P}} \right] \geq \lambda \right] \geq 1 - \delta \right\}.
$$

The proof of Lemma 5.7 immediately follows by the next lemma.

Lemma 5.11. *For every* k-*size subset* $\mathcal{P} \subset [n]$ *and* $d \in [h]$, *it holds that*

$$
\Pr_{D_{\mathcal{F}}} \left[\mathcal{T}_{d,b}^{\mathcal{P}} \right] \geq \gamma - 7\lambda - \frac{\chi + \alpha}{\lambda}.
$$

Proof of Lemma 5.7. Immediate by Lemma 5.11. □

The rest of this subsection is devoted to proving Lemma 5.11. Fix a k-size subset $\mathcal{P} \subset [n]$ and omit it from the notation when clear from the context. Let

$$\widetilde{\mathcal{V}}_{d,c} = \left\{ (f, \mathcal{S}, r) \colon \forall a \in \{0, 1\} \quad \Pi_{d+a}^{\mathcal{P}, \mathcal{S}}(v; (f, r))_{\overline{\mathcal{P} \cup \mathcal{S}}} = c^{\left|\overline{\mathcal{P} \cup \mathcal{S}}\right|} \right\}.$$

Namely, $\widetilde{\mathcal{V}}_{d,c} \subseteq \mathcal{V}_{d,c}$ are the sets, setup parameters and random strings, on which honest parties in $\Pi_{d+a}^{\mathcal{P}, \mathcal{S}}$ halt in the second round and output c, if the parties in $\mathcal{S}$ abort *and regardless* of whether the parties in $\mathcal{P}$ act toward those in $\mathcal{L}_{d+1}$ according to input 0 or 1. Let

$$\widetilde{\mathcal{T}}_{d,c} = \left\{ f \colon \Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\Pr_{r \leftarrow \mathcal{R}^n} \left[(f, \mathcal{S}, r), (f, \emptyset, r) \in \widetilde{\mathcal{V}}_{d,c} \right] \geq \lambda \right] \geq 1 - \delta \right\},$$

let $\widetilde{\mathcal{T}}_d = \widetilde{\mathcal{T}}_{d,0} \cup \widetilde{\mathcal{T}}_{d,1}$, and let $\widetilde{\mathcal{T}} = \bigcap_{d \in (h-1)} \widetilde{\mathcal{T}}_d$. Lemma 5.11 is proved via the following claims (the following probabilities are taken over $f \leftarrow D_{\mathcal{F}}$).

Claim 5.12. $\Pr\left[\mathcal{T}_{d+1,b} \mid \widetilde{\mathcal{T}} \right] < \eta$ *implies* $\Pr\left[\mathcal{T}_{d,1-b} \mid \widetilde{\mathcal{T}} \right] \geq 1 - \eta$.

Proof of Claim 5.12.   Assuming $\Pr\left[\mathcal{T}_{d+1,b} \mid \widetilde{\mathcal{T}} \right] \leq \eta$ notice that

$$\Pr\left[\widetilde{\mathcal{T}}_{d,b} \mid \widetilde{\mathcal{T}} \right] \leq \Pr\left[\mathcal{T}_{d+1,b} \mid \widetilde{\mathcal{T}} \right] \leq \eta.$$

Consequently, since $\Pr\left[\widetilde{\mathcal{T}}_d \mid \widetilde{\mathcal{T}} \right] = 1$, it follows that $\Pr\left[\widetilde{\mathcal{T}}_{d,b} \mid \widetilde{\mathcal{T}} \right] \leq \eta$ implies $\Pr\left[\widetilde{\mathcal{T}}_{d,1-b} \mid \widetilde{\mathcal{T}} \right] \geq 1 - \eta$ and thus $\Pr\left[\mathcal{T}_{d,1-b} \mid \widetilde{\mathcal{T}} \right] \geq \Pr\left[\widetilde{\mathcal{T}}_{d,1-b} \mid \widetilde{\mathcal{T}} \right] \geq 1 - \eta$. $\square$

Claim 5.13. $\Pr\left[\widetilde{\mathcal{T}} \right] \geq \gamma - 5\lambda$.

Claim 5.14. $\Pr\left[\mathcal{T}_{1,b} \mid \widetilde{\mathcal{T}} \right] \geq 1 - (\chi + \alpha)/(\Pr[\widetilde{\mathcal{T}}] \cdot \lambda)$.

Claim 5.15. *For every $d \in [h-1]$.*

$$\Pr\left[\mathcal{T}_{d,0} \mid \widetilde{\mathcal{T}} \right] + \Pr\left[\mathcal{T}_{d,1} \mid \widetilde{\mathcal{T}} \right] \leq 1 + \frac{\lambda}{h \cdot \Pr[\widetilde{\mathcal{T}}]}.$$

We prove Claims 5.13 to 5.15 below, but first use the above claims for proving Lemma 5.7.
Proving Lemma 5.11.

Proof of Lemma 5.11.   We first prove that for every $d \in [h]$:

$$\Pr\left[\mathcal{T}_{d,b} \mid \widetilde{\mathcal{T}} \right] \geq 1 - \frac{\chi + \alpha}{\Pr[\widetilde{\mathcal{T}}] \cdot \lambda} - \frac{d\lambda}{h \cdot \Pr[\widetilde{\mathcal{T}}]} \tag{22}$$

The proof is by induction on d. The base case, $d = 1$, is by Claim 5.14. The induction steps follows by the combination of Claim 5.15 and the contrapositive of Claim 5.12. Applying Eq. (22) for $d = h$, yields that

$$\Pr\left[\mathcal{T}_{h,b}\right] \geq \Pr[\tilde{\mathcal{T}}] - \frac{\chi + \alpha}{\lambda} - \lambda,$$

and the proof follows by Claim 5.13.   □

So it is left to prove Claims 5.13 to 5.15. Note that the following adversaries corrupt at most $k + \ell + 2\sigma n \leq t$ parties and thus they make a valid attack. Since our security model considers rushing adversaries, and Π has public randomness, we assume the adversary knows $f = (f_1, \ldots, f_n)$ *before* sending its first-round messages. In the following we let $\Pi_d^{\mathcal{S}} = \Pi_d^{\mathcal{P}, \mathcal{S}}$ and $\Pi_d = \Pi_d^{\emptyset}$.
Proving Claim 5.13. This is the only part in proof where we exploit the fact that the protocol is secure against *adaptive* adversaries.

Proof of Claim 5.13.   For $d \in (h)$, let $\mathcal{V}_d^{\mathcal{P}} = \mathcal{V}_{d,0}^{\mathcal{P}} \cup \mathcal{V}_{d,1}^{\mathcal{P}}$ and $\tilde{\mathcal{V}}_d = \tilde{\mathcal{V}}_{d,0} \cup \tilde{\mathcal{V}}_{d,1}$. Since $\Pr_{r \leftarrow \mathcal{R}^n}\left[(f, \mathcal{S}, r), (f, \emptyset, r) \in \tilde{\mathcal{V}}_d\right] \leq \sum_{c \in \{0,1\}} \Pr_{r \leftarrow \mathcal{R}^n}\left[(f, \mathcal{S}, r), (f, \emptyset, r) \in \tilde{\mathcal{V}}_{d,c}\right]$, for $f \notin \tilde{\mathcal{T}}_d$ it holds that

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}}\left[\Pr_{r \leftarrow \mathcal{R}^n}\left[(f, \mathcal{S}, r), (f, \emptyset, r) \in \tilde{\mathcal{V}}_d\right] \geq 2\lambda\right] < \delta \qquad (23)$$

Consider the following *rushing adaptive* adversary.

Algorithm 5.16.   *(A)*

Pre interaction: *Corrupt the parties in $\mathcal{P}$.*
First round.   *Let f be the parties' setup parameters.*
　　Do $\lceil 1/\lambda\delta \rceil$ times:

　　　1. *Sample $\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}$ conditioned on $|\mathcal{S}| \leq 2\sigma n$.*
　　　2. *For each $i \in (h - 1)$: estimate $\xi_i = \Pr_{r \leftarrow \mathcal{R}^n}\left[(f, \mathcal{S}, r), (f, \emptyset, r) \in \tilde{\mathcal{V}}_i\right]$ by*
　　　　taking $\Theta(\log(h/\lambda))$ samples of r. Let ξ_i' be the result of this estimation.
　　　3. *Let $d = \arg\min_{i \in (h-1)}\left\{\xi_i'\right\}$.*
　　　4. *If $\xi_d' < 3\lambda$, break the loop.*

　　Corrupt the parties in $\mathcal{S} \cup \mathcal{L}_{d+1}$ ($\mathcal{S}$ is the set sampled in the last loop), and act
　　according to Π_d.
Second round.   *Let r be the parties' second-round randomness.*
　　If $(f, \mathcal{W}, r) \notin \mathcal{V}_{d+a}$ for some $a \in \{0, 1\}$ and $\mathcal{W} \in \{\emptyset, \mathcal{S}\}$,
　　　act according to $\Pi_{d+a}^{\mathcal{W}}$.
　　Else, abort.
- -

By definition, if the attack does not abort then it violates either agreement or (second-round) halting. Let D be the value of d chosen by the adversary A at the first round of

the protocol. By construction, the attack abort with probability ξ_D. So it is left to argue about the value of ξ_D.

Assume $f \notin \widetilde{\mathcal{T}}$. Recall that (by Chernoff/Hoeffding bound) $\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} [|\mathcal{S}| > 2\sigma n] \leq \alpha < \delta/2$. Therefore, with probability at least $\delta/2$ over the choice of $\mathcal{S}$ in Step 1 of A, there exists $i \in (h-1)$ such that $\xi_i < 2\lambda$ (by Eq. (23)). It follows that $\xi_D < 3\lambda$, except with probability at most λ (i.e., error estimating ξ_D by another Chernoff/Hoeffding bound). We conclude that if $f \notin \widetilde{\mathcal{T}}$ the attack succeeds with probability at least $1 - 4\lambda$.

It follows that under the above attack, the honest parties halt in the second round and output the same value with probability at most $\Pr[\widetilde{\mathcal{T}}] + \Pr[\neg\widetilde{\mathcal{T}}] \cdot 4\lambda \leq \Pr[\widetilde{\mathcal{T}}] + 4\lambda$. Since the parties halt *and* agree with probability at least $\gamma - \alpha$, we conclude that $\Pr[\widetilde{\mathcal{T}}] \geq \gamma - \alpha - 4\lambda \geq \gamma - 5\lambda$. □

Proving Claim 5.14.

Proof of Claim 5.14. By definition, for $f \in \mathcal{T}_{1,b}$ it holds that

$$\Pr_{r \leftarrow \mathcal{R}^n} \left[\Pi_1(v; (f, r))_{\overline{\mathcal{H}}} = \overline{b}^{|\overline{\mathcal{H}}|} \right] = \Pr_{r \leftarrow \mathcal{R}^n} \left[(f, \emptyset, r) \in \mathcal{V}_{1,\overline{b}} \right] \geq \lambda,$$

letting $\mathcal{H} = \mathcal{P} \cup \mathcal{L}_1$ and $\overline{\mathcal{H}} = [n] \setminus \mathcal{H}$. Let $\eta = \Pr_f \left[\mathcal{T}_{1,\overline{b}} \mid \widetilde{\mathcal{T}} \right]$, clearly, $\Pr_f \left[\mathcal{T}_{1,b} \mid \widetilde{\mathcal{T}} \right] = 1 - \eta$. By the above

$$\Pr \left[\Pi_1(v)_{\overline{\mathcal{H}}} = \overline{b}^{|\overline{\mathcal{H}}|} \right] \geq \Pr[\widetilde{\mathcal{T}}] \cdot \eta \cdot \lambda \tag{24}$$

(recall that $\Pi_1(v)$ stands for $\Pi_1(v; (f, r))$, for a random choice of (f, r))). Finally, we notice that

$$\Pr \left[\Pi_1(v) = \overline{b}^{|\overline{\mathcal{H}}|} \right] + \Pr \left[\Pi(v) = b^n \right] \leq 1 + \alpha \tag{25}$$

If not, then the following attack violates the α-agreement. Recall that Π is an honest execution on input v and Π_1 is an execution of the protocol where the parties in $\mathcal{L}_1$ receive inputs from $\mathcal{P}$ according to input v' and all others receive inputs from $\mathcal{P}$ according to input v (recall that v and v' differ on exactly those indices indexed by $\mathcal{P}$). The attack proceeds as follows: the adversary corrupts the parties in $\mathcal{H}$, partitions the honest parties into two equal-size sets and acts toward the first honest parties according to Π and toward the rest according to Π_1. We conclude that $\Pr[\widetilde{\mathcal{T}}] \cdot \eta \cdot \lambda \leq \chi + \alpha$, and therefore $\eta \leq (\chi + \alpha)/(\Pr[\widetilde{\mathcal{T}}] \cdot \lambda)$. □

Proving Claim 5.15. The proof uses Conjecture 3.8 in a similar way to the second part of the proof of the theorem.

Proof of Claim 5.15. For $r \in (\mathcal{R} \cup \{\bot\})^n$ let $\mathcal{E}(r)$ be the indices in r of the value $\bot$. We assume without loss of generality that a party aborts upon getting $\bot$ as its second-round

random coins. For $f \in \text{Supp}(D_{\mathcal{F}})$, for $d \in [h-1]$, and for $b \in \{0, 1\}$, let

$$\mathcal{A}_b^f = \left\{ \boldsymbol{r} \in \{\mathcal{R} \cup \{\bot\}\} : \Pi_d(\boldsymbol{v}; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{E}(\boldsymbol{r})}} = b^{\left| \overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{E}(\boldsymbol{r})} \right|} \right\}. \tag{26}$$

By definition, for $f \in \mathcal{T}_{d,0} \cap \mathcal{T}_{d,1}$ and $b \in \{0, 1\}$, it holds that

$$\Pr_{\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\Pr_{\boldsymbol{r} \leftarrow \mathcal{R}^n} \left[\boldsymbol{r}, \bot_{\mathcal{S}}(\boldsymbol{r}) \in \mathcal{A}_b^f \right] \geq \lambda \right] \geq 1 - \delta. \tag{27}$$

By Conjecture 3.8, see Eq. (16), for $f \in \mathcal{T}_{d,0} \cap \mathcal{T}_{d,1}$ it holds that

$$\Pr_{\boldsymbol{r} \leftarrow \mathcal{R}^n, \mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\forall b \in \{0, 1\}: \{\boldsymbol{r}, \bot_{\mathcal{S}}(\boldsymbol{r})\} \cap \mathcal{A}_b^f \neq \emptyset \right] > \delta.$$

That is,

$$\Pr_{\boldsymbol{r} \leftarrow \mathcal{R}^n, \mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\forall b \in \{0, 1\} \ \exists \mathcal{S}_b \in \{\mathcal{S}, \emptyset\} : \Pi_d^{\mathcal{S}_b}(\boldsymbol{v}; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{S}_b}} = b^{\left| \overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{S}_b} \right|} \right]$$
$$> \delta. \tag{28}$$

In pursuit of contradiction, assume that $\Pr\left[\mathcal{T}_{d,0} \mid \widetilde{\mathcal{T}}\right] + \Pr\left[\mathcal{T}_{d,1} \mid \widetilde{\mathcal{T}}\right] \geq 1 + \lambda/(h \cdot \Pr[\widetilde{\mathcal{T}}])$ for some $d \in [h-1]$. It follows that

$$\Pr_{\substack{f \leftarrow D_{\mathcal{F}} \\ \boldsymbol{r} \leftarrow \mathcal{R}^n, \mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}}} \left[\forall b \in \{0, 1\} \ \exists \mathcal{S}_b \in \{\mathcal{S}, \emptyset\} : \Pi_d^{\mathcal{S}_b}(\boldsymbol{v}; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{S}_b}} = b^{\left| \overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{S}_b} \right|} \right]$$
$$> \Pr\left[\mathcal{T}_{d,0} \cap \mathcal{T}_{d,1}\right] \cdot \delta$$
$$\geq \Pr[\widetilde{\mathcal{T}}] \cdot \Pr[\mathcal{T}_{d,0} \cap \mathcal{T}_{d,1} \mid \widetilde{\mathcal{T}}] \cdot \delta$$
$$\geq \Pr[\widetilde{\mathcal{T}}] \cdot \frac{\lambda}{h \cdot \Pr[\widetilde{\mathcal{T}}]} \cdot \delta$$
$$= \lambda \delta / h$$
$$> 8\alpha. \tag{29}$$

The first inequality is by Eq. (28), the second one by the assumption that $\Pr[\mathcal{T}_{d,0} \mid \widetilde{\mathcal{T}}] + \Pr[\mathcal{T}_{d,1} \mid \widetilde{\mathcal{T}}] \geq 1 + \lambda/(h \cdot \Pr[\widetilde{\mathcal{T}}])$, and the last one by the definition of α. Next, consider the following rushing adversary:

Algorithm 5.17. *(A)*

Pre-interaction.

    1. For each $i \in [h-1]$, estimate

$$\xi_i = \Pr_{\boldsymbol{r} \leftarrow \mathcal{R}^n, \mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}} \left[\forall b \in \{0, 1\} \ \exists \mathcal{S}_b \in \{\mathcal{S}, \emptyset\} : \Pi_d^{\mathcal{S}_b}(\boldsymbol{v}; (\boldsymbol{f}, \boldsymbol{r}))_{\overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{S}_b}} = b^{\left| \overline{\mathcal{P} \cup \mathcal{L}_d \cup \mathcal{S}_b} \right|} \right]$$

by taking $\Theta(\log(h/\alpha))$ samples. Let $d = \text{argmax}_{i \in [h-1]} \{\xi_i\}$.

2. *Sample a random $\mathcal{S} \leftarrow \mathbf{D}_{n,\sigma}$ conditioned on $|\mathcal{S}| \leq 2\sigma n$.*

Corrupt the parties in $\mathcal{P} \cup \mathcal{S} \cup \mathcal{L}_d$.

First round. *Act according to Π_d.*

Second round. *Partition the honest parties arbitrarily into two equal-size sets $\mathcal{H}_1$ and $\mathcal{H}_2$, and act towards $\mathcal{H}_1$ according to $\Pi_d^{\mathcal{S}}$ and towards $\mathcal{H}_2$ according to $\Pi_d^{\emptyset}$.*

Observe that Eq. (29) says that with probability 8α (over the setup parameter, the choice of set S and coins r) the output of the honest parties is sensitive to whether the parties in S abort or not (while halting and agreement occurs for both cases). Therefore, analogously to the proof of Claim 5.13, we deduce that the adversary described above causes disagreement with probability at least α. $\qquad\square$

Acknowledgements

A. Locally Consistent Security to Malicious Security

In this section, we formally state and prove Theorem 1.6 and show how to *compile* any BA protocol that is secure against locally consistent adversaries into a protocol that is secure against malicious adversaries. That is, we prove the following theorem:

Theorem A.1. (Theorem 1.6, restated) *Let Π be a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent adversaries for $q = O(\log n)$ and assume the existence of verifiable random functions and existentially unforgeable digital signatures under an adaptive chosen-message attack. Then,*

1. *Assuming in addition the existence of non-interactive zero-knowledge proofs, there exist a PPT protocol-compiler $\mathsf{Comp}(\cdot)$ such that $\Pi' = \mathsf{Comp}(\Pi)$ is a $(t, \alpha - \text{neg}(\kappa), \beta - \text{neg}(\kappa), q, \gamma - \text{neg}(\kappa))$-BA in the PKI model, resilient to malicious adversaries.*
2. *There exists a PPT protocol-compiler $\mathsf{Comp}_{\mathsf{PR}}(\cdot)$ such that if Π is a public-randomness protocol, then $\Pi' = \mathsf{Comp}_{\mathsf{PR}}(\Pi)$ is a $(t, \alpha - \text{neg}(\kappa), \beta - \text{neg}(\kappa), q, \gamma - \text{neg}(\kappa))$-BA in the PKI model, resilient to malicious adversaries.*

In Sect. A.1, we define the cryptographic primitives used in the compiler, and in Sect. A.2, we construct the compiler and prove its security.

A.1. *Preliminaries*

The compiler makes use of *verifiable random functions* (VRF) [56], *digital signatures*, and non-interactive zero-knowledge proofs, as defined below.

A.1.1. *Verifiable Random Functions*

We follow the definition of VRF from [43].

Definition A.2. (VRF) A verifiable random function is a tuple of polynomial-time algorithms $\Pi = (\mathsf{VRF.Gen}, \mathsf{VRF.Eval}, \mathsf{VRF.Verify})$ of the following form.

- $\mathsf{VRF.Gen}(1^\kappa) \to (sk, vk)$. On input the security parameter, the key-generation algorithm outputs a secret key sk and a public verification key vk.
- $\mathsf{VRF.Eval}(sk, x) \to (y, \pi)$. On input the secret key and an input $x \in \{0, 1\}^\kappa$, the evaluation algorithm outputs a value $y \in \mathcal{S}$ (for a finite set $\mathcal{S}$) and a proof π.
- $\mathsf{VRF.Verify}(vk, x, y, \pi) \to b$. On input the verification key, an input $x \in \{0, 1\}^\kappa$, an output $y \in \mathcal{S}$, and a proof π, the deterministic verification algorithm outputs a bit $b \in \{0, 1\}$.

We require the following properties:

- **Correctness.** For $(sk, vk) \leftarrow \mathsf{VRF.Gen}(1^\kappa)$ and $x \in \{0, 1\}^\kappa$ it holds that if $(y, \pi) \leftarrow \mathsf{VRF.Eval}(sk, x)$ then $\mathsf{VRF.Verify}(vk, x, y, \pi) = 1$.
- **Unique provability.** For all strings (sk, vk) (not necessarily generated by $\mathsf{VRF.Gen}$) and all $x \in \{0, 1\}^\kappa$, there exists no (y_0, π_0, y_1, π_1) such that $y_0 \neq y_1$ and $\mathsf{VRF.Verify}(vk, x, y_0, \pi_0) = \mathsf{VRF.Verify}(vk, x, y_1, \pi_1) = 1$.
- **Pseudorandomness.** For any PPT adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ it holds that

$$\left| \Pr\left[\mathsf{Expt}_{\Pi,\mathsf{A}}^{\mathsf{VRF}}(\kappa) = 1 \right] - \frac{1}{2} \right| \leq \mathrm{neg}(\kappa),$$

for the experiment $\mathsf{Expt}^{\mathsf{VRF}}$ defined below:

$\mathsf{Expt}_{\Pi,\mathsf{A}}^{\mathsf{VRF}}(\kappa)$	$\mathcal{O}_{\mathsf{eval}}(x)$
$(sk, vk) \leftarrow \mathsf{VRF.Gen}(1^\kappa)$	$(y, \pi) \leftarrow \mathsf{VRF.Eval}(sk, x)$ return (y, π)
$(x^*, \mathsf{state}) \leftarrow \mathsf{A}_1^{\mathcal{O}_{\mathsf{eval}}(\cdot)}(vk)$	
$(y_0, \pi) \leftarrow \mathsf{VRF.Eval}(sk, x^*)$	
$y_1 \leftarrow_R \mathcal{S}$	
$b \leftarrow_R \{0, 1\}$	
$b' \leftarrow \mathsf{A}_2^{\mathcal{O}_{\mathsf{eval}}(\cdot)}(\mathsf{state}, y_b)$	
return 1 if and only if $b = b'$	
and A didn't query x^*	

A.1.2. *Digital Signatures*

We consider the standard notion of existentially unforgeable signatures under an adaptive chosen-message attack [38].

Definition A.3. (*Digital signatures*) A digital signatures scheme is a tuple of polynomial-time algorithms $\Pi = (\mathsf{DS.Gen}, \mathsf{DS.Sign}, \mathsf{DS.Verify})$ of the following form.

- $\mathsf{DS.Gen}(1^\kappa) \to (sk, vk)$. On input the security parameter, the key-generation algorithm outputs a secret signing key sk and a public verification key vk.
- $\mathsf{DS.Sign}(sk, m) \to \sigma$. On input the signing key and a message m, the signing algorithm outputs a signature σ.
- $\mathsf{DS.Verify}(vk, m, \sigma) \to b$. On input the verification key, a message m, and a signature σ, the deterministic verification algorithm outputs a bit $b \in \{0, 1\}$.

We require the following properties:

- **Correctness.** For $(sk, vk) \leftarrow \mathsf{DS.Gen}(1^\kappa)$ and a message m it holds that if $\sigma \leftarrow \mathsf{DS.Sign}(sk, m)$ then $\mathsf{DS.Verify}(vk, m, \sigma) = 1$.
- **Existentially unforgeable under an adaptive chosen-message attack.** For any PPT adversary A it holds that

$$\left| \Pr\left[\mathsf{Expt}^{\mathsf{Sig}}_{\Pi, \mathsf{A}}(\kappa) = 1 \right] \right| \leq \mathsf{neg}(\kappa),$$

for the experiment $\mathsf{Expt}^{\mathsf{Sig}}$ defined below:

$\mathsf{Expt}^{\mathsf{Sig}}_{\Pi, \mathsf{A}}(\kappa)$	$\mathcal{O}_{\mathsf{sign}}(m)$
$(sk, vk) \leftarrow \mathsf{DS.Gen}(1^\kappa)$	$\sigma \leftarrow \mathsf{DS.Sign}(sk, m)$ return σ
$(m, \sigma) \leftarrow \mathsf{A}^{\mathcal{O}_{\mathsf{sign}}(\cdot)}(vk)$	
return 1 if and only if $\mathsf{DS.Verify}(vk, m, \sigma) = 1$	
and A didn't query m	

A.1.3. *Non-interactive Zero-Knowledge Proofs*

A non-interactive zero-knowledge proof [11] is a single-message protocol that allow a prover to convince a verifier the a certain common statement belongs to a language, without disclosing any additional information. We follow the definition from [41].

Definition A.4. (NIZK) Let $\mathcal{R}$ be an NP-relation and let $\mathcal{L}_\mathcal{R}$ be the language consisting of the statements in $\mathcal{R}$. A non-interactive zero-knowledge proof system for $\mathcal{R}$ is a tuple of polynomial-time algorithms $\Pi = (\mathsf{NIZK.Gen}, \mathsf{NIZK.Prover}, \mathsf{NIZK.Verifier})$ of the following form:

- $\mathsf{NIZK.Gen}(1^\kappa) \to \mathsf{crs}$. On input the security parameter, the setup-generation algorithm outputs a common reference string crs.
- $\mathsf{NIZK.Prover}(\mathsf{crs}, x, w) \to \varphi$. On input the crs, a statement x, and a witness w such that $(x, w) \in \mathcal{R}$, the prover algorithm outputs a proof string φ.
- $\mathsf{NIZK.Verifier}(\mathsf{crs}, x, \varphi) \to b$. On input the crs, a statement x, and a proof φ, the verification algorithm outputs a bit $b \in \{0, 1\}$.

We require the following properties:

- **Correctness.** A proof system is complete if an honest prover with a valid witness can convince an honest verifier. For $(x, w) \in \mathcal{R}$ it holds that

$$\Pr\left[\mathsf{NIZK.Verifier}(\mathsf{crs}, x, \varphi) = 1 \mid \mathsf{crs} \leftarrow \mathsf{NIZK.Gen}(1^\kappa), \varphi \leftarrow \mathsf{NIZK.Prover}(\mathsf{crs}, x, w)\right] = 1.$$

- **Statistical soundness.** A proof system is sound if it is infeasible to convince an honest verifier when the statement is false. For all polynomial-size families $\{x_\kappa\}$ of statements $x_\kappa \notin \mathcal{L}_\mathcal{R}$ and all adversaries A it holds that

$$\Pr\left[[\mathsf{NIZK.Verifier}(\mathsf{crs}, x_\kappa, \varphi) = 1 \mid \mathsf{crs} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)\varphi \leftarrow \mathsf{A}(\mathsf{crs}, x_\kappa)] = 1.\right]$$

- **Computational (adaptive, multi-theorem) zero knowledge.** A proof system is zero-knowledge if the proofs do not reveal any information about the witnesses. There exists a polynomial-time simulator $\mathsf{S}_{\mathsf{nizk}}xspace = (\mathsf{S}_{\mathsf{nizk}}xspace^1, \mathsf{S}_{\mathsf{nizk}}xspace^2)$, where $\mathsf{S}_{\mathsf{nizk}}xspace^1$ returns a simulated crs together with a simulation trapdoor τ that enables $\mathsf{S}_{\mathsf{nizk}}xspace^2$ to simulate proofs without having access to the witness. That is, for every non-uniform polynomial-time adversary A it holds that

$$\left| \Pr\left[\mathsf{A}^{\mathsf{P}_{\mathsf{crs}}(\cdot,\cdot)}(\mathsf{crs}) = 1 \mid \mathsf{crs} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)\right] \right.$$
$$\left. - \Pr\left[\mathsf{A}^{\mathsf{S}_{\mathsf{crs},\tau}(\cdot,\cdot)}(\mathsf{crs}) = 1 \mid (\mathsf{crs}, \tau) \leftarrow \mathsf{S}_{\mathsf{nizk}}xspace^1(1^\kappa)\right] \right|$$
$$\leq \mathsf{neg}(\kappa),$$

where $\mathsf{S}_{\mathsf{crs},\tau}(x, w) = \mathsf{S}_{\mathsf{nizk}}xspace^2(\mathsf{crs}, \tau, x)$ for $(x, w) \in \mathcal{R}$ and $\mathsf{P}_{\mathsf{crs}}(x, w) = \mathsf{NIZK.Prover}(\mathsf{crs}, x, w)$.

A.1.4. *Next-Message Functions*

An n-party protocol is represented by a set $\{\mathsf{next\text{-}msg}_{i \to j}\}_{i,j \in [n]}$ of next-message functions, a set $\{\mathsf{output}_i\}_{i \in [n]}$ of output functions, and a distribution D for generating setup information. Initially, the setup information is sampled as $(\mathsf{setup}_1, \ldots, \mathsf{setup}_n) \leftarrow D$ and every party P_i receives setup_i before the protocol begins. The view of a party P_i in the r'th round, denoted view_i^r, consists of: its input bit x_i, its setup information setup_i, its random coin tosses $\rho_i = (\rho_i^1, \ldots, \rho_i^r)$ (where $\rho_i^{r'}$ are the tossed coins for round r') and the incoming messages $(m_{1 \to i}^{r'}, \ldots, m_{n \to i}^{r'})$ for every $r' < r$, where $m_{j \to i}^{r'}$ is the message received from P_j in round r'. Given P_i's view in the r'th round, the function $\mathsf{next\text{-}msg}_{i \to j}(\mathsf{view}_i^r)$ outputs the message $m_{i \to j}^r$ to be sent by P_i to P_j, except for the last round, where it outputs $\bot$; in that case the output function $\mathsf{output}(\mathsf{view}_i^r)$ produces the output value y. Without loss of generality we assume that a message $m_{i \to j}^r$ is of the form (r, i, j, m); looking ahead, this will ensure that two messages in the protocol will not have the same signature.

A.1.5. *The PKI Model*

The compiled protocol is designed to work in the *public-key infrastructure* (PKI) model, where a trusted third party generates private/public keys for the parties before the protocol begins. In our setting, we will require a PKI for VRF, digital signatures, and NIZK, meaning that the trusted party operates as follows:

1. For every $i \in [n]$, compute VRF keys $(\mathsf{sk}_i^{\mathsf{vrf}}, \mathsf{vk}_i^{\mathsf{vrf}}) \leftarrow \mathsf{VRF.Gen}(1^\kappa)$.
2. For every $i \in [n]$, compute signature keys $(\mathsf{sk}_i^{\mathsf{ds}}, \mathsf{vk}_i^{\mathsf{ds}}) \leftarrow \mathsf{DS.Gen}(1^\kappa)$.
3. Compute $\mathsf{crs} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$.
4. Send to every party P_i the secret keys $(\mathsf{sk}_i^{\mathsf{vrf}}, \mathsf{sk}_i^{\mathsf{ds}})$ as well as all the public keys $\mathsf{crs}, (\mathsf{vk}_1^{\mathsf{vrf}}, \ldots, \mathsf{vk}_n^{\mathsf{vrf}})$ and $(\mathsf{vk}_1^{\mathsf{ds}}, \ldots, \mathsf{vk}_n^{\mathsf{ds}})$.

A.2. *The Compiler*

Given a protocol that is secure against locally consistent adversaries, the main idea of the compiler is to limit the capabilities of a malicious adversary attacking the compiled protocol to those of a locally consistent one. This is achieved by proving an honest behavior via the cryptographic tools described above (VRF, digital signatures, and NIZK proofs) in a similar way to the GMW compiler [36]. Unlike GMW, where all consistency proofs are carried out over a broadcast channel to ensure a consistent view between the honest parties, in our case the consistency proofs are done over pairwise channels, so they only guarantee local consistency.

We start by defining the NP relations that will be used for the zero-knowledge proofs. Each instance consists of a message between a pair of parties (say from P_i' to P_j') and the witness is the internal state of P_i' used to generate the message (the input, the random coins, and all incoming messages) along with a "proof of correctness," i.e., that the random coins were properly generated using the VRF, that the incoming messages that P_i' received from every P_k' were signed by P_k', and in turn were proven to be generated correctly (i.e., that each P_k' used the correct random coins generated by the VRF and its incoming messages were signed by the senders). Note that this recursive step in the verification is required for proving locally consistent behaviour, since if both P_i' and P_k' are corrupt, then P_k' can send an arbitrary message to P_i' and sign it (in this case the NIZK proof from P_k' to P_i' will not verify). When P_i' sends its message to an honest P_j', it is not enough that P_i' proves that the messages from P_k' are properly signed, but P_i' must also prove that P_k' provided a NIZK proof asserting that its messages were generated by consistent random coins and correct incoming messages according to the next-message function. For this reason we consider $q = O(\log n)$

The Relation $\mathcal{R}_{i \to j}^r$. We will consider the following set of NP relations, where for $i, j \in [n]$ and an integer r, the relation $\mathcal{R}_{i \to j}^r$ is parametrized by an n-party protocol Π (represented by $\{\mathsf{next\text{-}msg}_{i \to j}\}_{i,j \in [n]}$ and $\{\mathsf{output}_i\}_{i \in [n]}$), a VRF scheme, a DS scheme, and a NIZK scheme, as well as:

- A vector of VRF verification keys $(\mathsf{vk}_1^{\mathsf{vrf}}, \ldots, \mathsf{vk}_n^{\mathsf{vrf}})$.
- A vector of signature verification keys $(\mathsf{vk}_1^{\mathsf{ds}}, \ldots, \mathsf{vk}_n^{\mathsf{ds}})$.
- A NIZK common reference string crs.

The instance consists of a message $(m_{i\to j}^r, \sigma_{i\to j}^r, \pi_i^r)$ (the message from P_i to P_j). The witness consists of:

- A bit $x_i \in \{0, 1\}$ and a string setup_i.
- A vector of random coins $(\rho_i^1, \ldots, \rho_i^r)$.
- For $r' \in [r-1]$ and $k \in [n]$, a message $\boldsymbol{m}_{k\to i}^{r'} = (m_{k\to i}^{r'}, \sigma_{k\to i}^{r'}, \pi_k^{r'}, \varphi_{k\to i}^{r'})$ (P_i's incoming messages).

The instance/witness pair is in the relation $\mathcal{R}_{i\to j}^r$ if the following holds:

1. For every $r' \in [r]$ it holds that $\mathsf{VRF.Verify}(\mathsf{vk}_i^{\mathsf{vrf}}, (i, r'), \rho_i^{r'}, \pi_i^{r'}) = 1$.
2. $\mathsf{DS.Verify}(\mathsf{vk}_i^{\mathsf{ds}}, m_{i\to j}^r, \sigma_{i\to j}^r) = 1$.
3. For $r' \in [r-1]$ and $k \in [n]$ it holds that $\mathsf{NIZK.Verifier}(\mathsf{crs}, (m_{k\to i}^{r'}, \sigma_{k\to i}^{r'}, \pi_k^{r'}), \varphi_{k\to i}^{r'}) = 1$ with respect to the relation $\mathcal{R}_{k\to i}^{r'}$.
4. Set $\mathsf{view}_i^1 = (x_i, \mathsf{setup}_i, \rho_i^1)$ and for $1 < r' \le r$ set $\mathsf{view}_i^{r'} = (\mathsf{view}_i^{r'-1}, m_{1\to i}^{r'-1}, \ldots, m_{n\to i}^{r'-1}, \rho_i^{r'})$. Then, it holds that $m_{i\to j}^r = \mathsf{next\text{-}msg}_{i\to j}(\mathsf{view}_i^r)$.

The compiled protocol. Having defined the relations $\{\mathcal{R}_{i\to j}^r\}$, we are ready to present the compiler for a protocol Π, secure against locally consistent adversaries to a maliciously secure one. Initially, in the setup phase, each party receives its setup information for Π in addition to the PKI keys for VRF, digital signatures, and NIZK (as described above). To generate its coins for the r'th round (along with a proof), party P_i evaluates the VRF over the pair (i, r); next, P_i computes the r'th round messages for Π, signs each message, and sends to every other P_j the corresponding message, the signature, and the VRF proof. In addition, P_i sends to P_j a NIZK proof for $\mathcal{R}_{i\to j}^r$, proving that P_i behaves consistently towards P_j.

Let $\Pi = (P_1, \ldots, P_n)$ be an n-party protocol represented by the set of next-message functions $\{\mathsf{next\text{-}msg}_{i\to j}\}_{i,j\in[n]}$, the set of output functions $\{\mathsf{output}_i\}_{i\in[n]}$, and a distribution D for generating setup information. Let VRF be a verifiable random function, let DS be a digital signatures scheme, and let NIZK be a non-interactive zero-knowledge proof scheme. Later on, we will simplify the compiler for the case of public-randomness protocols by removing the need for NIZK.

Protocol A.5. *(Protocol $\Pi' = (P_1', \ldots, P_n') = \mathsf{Comp}(\Pi)$)*

Setup: *The setup-generation algorithm samples* $(\mathsf{setup}_1, \ldots, \mathsf{setup}_n) \leftarrow D$ *for the protocol* Π, *computes* $\mathsf{crs} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$, *and for every* $i \in [n]$ *computes* $(\mathsf{sk}_i^{\mathsf{vrf}}, \mathsf{vk}_i^{\mathsf{vrf}}) \leftarrow \mathsf{VRF.Gen}(1^\kappa)$ *and* $(\mathsf{sk}_i^{\mathsf{ds}}, \mathsf{vk}_i^{\mathsf{ds}}) \leftarrow \mathsf{DS.Gen}(1^\kappa)$. *The setup string for party* P_i' *is set to be* $\mathsf{setup}_i' = \big(\mathsf{setup}_i, \mathsf{sk}_i^{\mathsf{vrf}}, \mathsf{sk}_i^{\mathsf{ds}}, \mathsf{crs}, \mathsf{vk}_1^{\mathsf{vrf}}, \ldots,$ $\mathsf{vk}_n^{\mathsf{vrf}}, \mathsf{vk}_1^{\mathsf{ds}}, \ldots, \mathsf{vk}_n^{\mathsf{ds}}\big)$.

Input: *Party* P_i' *starts with an input bit* $x_i \in \{0, 1\}$.

Round $r = 1$:

1. P_i' *computes* $(\rho_i^1, \pi_i^1) \leftarrow \mathsf{VRF.Eval}(\mathsf{sk}_i^{\mathsf{vrf}}, (i, 1))$ *and sets* $\mathsf{view}_i^1 = (x_i, \mathsf{setup}_i, \rho_i^1)$.
2. P_i' *computes for every* $j \in [n]$ *the message* $m_{i\to j}^1 = \mathsf{next\text{-}msg}_{i\to j}(\mathsf{view}_i^1)$ *and signs* $\sigma_{i\to j}^1 \leftarrow \mathsf{DS.Sign}(\mathsf{sk}_i^{\mathsf{ds}}, m_{i\to j}^1)$.

3. P'_i computes for every $j \in [n]$ a proof for the relation $\mathcal{R}^1_{i \to j}$ on $\text{stat}^1_{i \to j} = (m^1_{i \to j}, \sigma^1_{i \to j}, \pi^1_i)$ and witness $\text{wit}^1_{i \to j} = (x_i, \text{setup}_i, \rho^1_i)$ as $\varphi^1_{i \to j} \leftarrow \text{NIZK.Prover}(\text{crs}, \text{stat}^1_{i \to j}, \text{wit}^1_{i \to j})$.

4. P'_i sends $\boldsymbol{m}^1_{i \to j} = (m^1_{i \to j}, \sigma^1_{i \to j}, \pi^1_i, \varphi^1_{i \to j})$ to P'_j.

Round $r > 1$: Let $\boldsymbol{m}^{r-1}_{j \to i} = (m^{r-1}_{j \to i}, \sigma^{r-1}_{j \to i}, \pi^{r-1}_j, \varphi^{r-1}_{i \to j})$ be the message P'_i received from P'_j in round $r-1$. If P'_j did not send a message, or if $\text{NIZK.Verifier}(\text{crs}, (m^{r-1}_{j \to i}, \sigma^{r-1}_{j \to i}, \pi^{r-1}_j), \varphi^{r-1}_{i \to j}) = 0$, set $m^{r-1}_{j \to i} = \bot$.

1. P'_i computes $(\rho^r_i, \pi^r_i) \leftarrow \text{VRF.Eval}(\text{sk}^{\text{vrf}}_i, (i, r))$ and sets the internal view as $\text{view}^r_i = (\text{view}^{r-1}_i, m^{r-1}_{1 \to i}, \ldots, m^{r-1}_{n \to i}, \rho^r_i)$.

2. P'_i computes for every $j \in [n]$ the message $m^r_{i \to j} = \text{next-msg}_{i \to j}(\text{view}^r_i)$ and signs $\sigma^r_{i \to j} \leftarrow \text{DS.Sign}(\text{sk}^{\text{ds}}_i, m^r_{i \to j})$.

3. P'_i computes for every $j \in [n]$ a proof for the relation $\mathcal{R}^r_{i \to j}$ on the statement $\text{stat}^r_{i \to j} = (m^r_{i \to j}, \sigma^r_{i \to j}, \pi^r_i)$ and witness $\text{wit}^r_{i \to j} = (\text{wit}^{r-1}_{i \to j}, \rho^r_i, \{m^{r-1}_{k \to i}\}_{k \in [n]})$ as $\varphi^r_{i \to j} \leftarrow \text{NIZK.Prover}(\text{crs}, \text{stat}^r_{i \to j}, \text{wit}^r_{i \to j})$.

4. P'_i sends $\boldsymbol{m}^r_{i \to j} = (m^r_{i \to j}, \sigma^r_{i \to j}, \varphi^r_{i \to j}, \pi^r_i)$ to P'_j.

Output: If in some round r, the output of $\text{next-msg}_{i \to j}(\text{view}^r_i)$ is $\bot$ for all $j \in [n]$, indicating it is the last round, P'_i outputs $y = \text{output}(\text{view}^r_i)$ and halts.

A.2.1. *Security Proof*

We prove the security of Protocol A.5 using a sequence of arguments. Given a protocol Π secure against locally consistent adversaries, we first adjust it to use pseudorandom coins computed using a VRF. The new protocol, denoted Π_1, remains secure against slightly weaker locally consistent adversaries by the pseudorandomness property of the VRF. Next, we show how to convert any malicious adversary against the compiled protocol $\Pi' = \text{Comp}(\Pi)$ into a "weak" locally consistent attack against Π_1. The proof of the second part of the theorem, concerning public-randomness protocols, follows in similar lines.

Proof of Theorem A.1. We start by proving the first part of the theorem, considering generic protocols, and later focus on public-randomness protocols.

Proof of Item 1 (generic protocols). We prove Item 1 in two steps. Initially, as an intermediate step, we consider a variant of Π, denoted Π_1, where the parties behave exactly as in Π except that they use a VRF to compute their random coins for each round. Formally, Π_1 is defined in the PKI model, where, in addition to the setup information for Π, every party P_i receives sk^{vrf}_i and $(\text{vk}^{\text{vrf}}_1, \ldots, \text{vk}^{\text{vrf}}_n)$ for $(\text{sk}^{\text{vrf}}_i, \text{vk}^{\text{vrf}}_i) \leftarrow \text{VRF.Gen}(1^\kappa)$. During the execution of the protocol, each party P_i evaluates $(\rho^r_i, \pi^r_i) \leftarrow \text{VRF.Eval}(\text{sk}^{\text{vrf}}_i, (i, r))$, sets its coins for the r'th round to ρ^r_i (instead of a uniformly distributed string), and appends π^r_i to its r'th round messages. Note that the strings ρ^r_i

are deterministic, so a locally consistent adversary has the power to use arbitrary values instead. To enable a reduction to the security of Π, we will explicitly assume that corrupted parties indeed use the honestly generated pseudorandom values ρ_i^r by evaluating the VRF on (i, r); we call such a locally consistent adversary VRF *-compliant*.

Claim A.6. *If Π is a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent adversaries, then Π_1 is a $(t, \alpha - \text{neg}(\kappa), \beta - \text{neg}(\kappa), q, \gamma - \text{neg}(\kappa))$-BA against locally consistent VRF-compliant adversaries.*

Proof. By assumption, a corrupted P_i uses the value ρ_i^r as its random coins for the r'th round. Therefore, the only difference between Π_1 and Π are the use of pseudorandom string instead of uniformly distributed strings. The proof follows by the pseudorandomness of the VRF scheme using a standard hybrid argument. $\square$

Next, let A' be an adversary attacking $\Pi' = (P_1', \ldots, P_n')$. We will construct an adversary A for the protocol $\Pi_1 = (P_1, \ldots, P_n)$. Let $S_{\text{nizk}}xspace = (S_{\text{nizk}}xspace^1, S_{\text{nizk}}xspace^2)$ be the simulator that is guaranteed for the NIZK scheme. The adversary A runs internally a copy of A' and proceeds as follows:

- In the setup phase of Π_1, A receives the setup string $\left(\text{setup}_i, \text{sk}_i^{\text{vrf}}, \text{vk}_1^{\text{vrf}}, \ldots, \text{vk}_n^{\text{vrf}}\right)$ (consisting of the setup for Π and the VRF keys). Next, A samples $(\text{crs}, \tau) \leftarrow S_{\text{nizk}}xspace^1(1^\kappa)$ and $(\text{sk}_i^{\text{ds}}, \text{vk}_i^{\text{ds}}) \leftarrow \text{DS.Gen}(1^\kappa)$ for every $i \in [n]$, and provides the setup string $\text{setup}_i' = \left(\text{setup}_i, \text{sk}_i^{\text{vrf}}, \text{sk}_i^{\text{ds}}, \text{crs}, \text{vk}_1^{\text{vrf}}, \ldots, \text{vk}_n^{\text{vrf}}, \text{vk}_1^{\text{ds}}, \ldots, \text{vk}_n^{\text{ds}}\right)$ for every corrupted P_i'.
- Upon receiving a message $(m_{i \to j}^r, \pi_i^r)$ from an honest P_i to a corrupted P_j in the execution of Π_1, A sends $(m_{i \to j}^r, \sigma_{i \to j}^r, \pi_i^r, \varphi_{i \to j}^r)$ to A' with $\sigma_{i \to j}^r \leftarrow \text{DS.Sign}(\text{sk}_i^{\text{ds}}, m_{i \to j}^r)$ and $\varphi_{i \to j}^r \leftarrow S_{\text{nizk}}xspace^2(\text{crs}, \tau, (m_{i \to j}^r, \sigma_{i \to j}^r, \pi_i^r))$.
- When A receives $(m_{i \to j}^r, \sigma_{i \to j}^r, \pi_i^r, \varphi_{i \to j}^r)$ from A' on behalf of a corrupted P_i' to an honest P_j' (in the simulated execution of Π'), A first verifies that $\text{NIZK.Verifier}(\text{crs}, (m_{i \to j}^r, \sigma_{i \to j}^r, \pi_i^r), \varphi_{i \to j}^r) = 1$. If the proof is verified, A sends the message $(m_{i \to j}^r, \pi_i^r)$ to P_j in the protocol Π_1; otherwise, A considers P_i as an aborting party towards P_j.

We complete the proof in a series of steps, analyzing the attack under increasingly stronger power of the adversary A', starting from a locally consistent VRF-compliant attack until reaching a full blown malicious attack. Initially, we will assume perfect security of the NIZK, and remove this restriction later on.

Claim A.7. *Consider a perfect NIZK scheme. If Π_1 is a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent VRF-compliant adversaries, then Π' is a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent VRF-compliant adversaries.*

Proof. If A' is a locally consistent VRF-compliant adversary, then in particular whenever A' sends a message on behalf of a corrupted P_i', he knows a witness for the NIZK

proof. Therefore, without loss of generality we can assume that either a corrupted P'_i does not send a message (i.e., aborts) to an honest P'_j or that P'_i correctly generates the NIZK proof. In that case every locally consistent VRF-compliant attack by A' translates to a locally consistent VRF-compliant attack by A. □

The next claim considers stronger adversaries that are allowed to use arbitrary random coins for computing the next-message function. We will use the following notations: A message sent in Π' is of the form $(m, \sigma, \pi, \varphi)$; we call m the *content of the message*. For a party P'_i, let $\mathcal{M}^{r',k \to i}_{in}$ denote the set of incoming messages' contents received from party P'_k in round r' (as this is a locally consistent attack, there could be multiple incoming messages from each corrupted party, but at most one message from each honest party). Let $\mathcal{M}^{r,i \to j}_{out}$ be the set of possible messages' contents that P'_i can send to P'_j at round r under a VRF-compliant locally consistent attack when using a subset of the incoming messages' contents $\{\mathcal{M}^{r',k \to i}_{in}\}_{r' < r, k \in [n]}$ and randomness $\{\rho^{r'}_i\}_{r' \in [r]}$ computed as $(\rho^{r'}_i, \pi^{r'}_i) \leftarrow \text{VRF.Eval}(sk^{vrf}_i, (i, r'))$.

Claim A.8. *Consider a perfect NIZK scheme. If Π_1 is a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent VRF-compliant adversaries, then Π' is a $(t, \alpha - \text{neg}(\kappa), \beta - \text{neg}(\kappa), q, \gamma - \text{neg}(\kappa))$-BA against locally consistent adversaries.*

Proof. We prove the claim by showing that the additional power of the adversary only allows for a negligible cheating advantage. Consider a locally consistent adversary A' and assume that a corrupted party P'_i used arbitrary random coins to generate the message content for party P'_j in round r, denoted $\tilde{m}^r_{i \to j}$. There are two possible cases:

Case 1: If $\tilde{m}^r_{i \to j} \in \mathcal{M}^{r,i \to j}_{out}$, then the adversary can compute a witness for the relation $\mathcal{R}^r_{i \to j}$. That is, even if the actual coins used to generate $\tilde{m}^r_{i \to j}$ are different than $\{\rho^{r'}_i\}_{r' \in [r]}$, the message $\tilde{m}^r_{i \to j}$ can be explained as if generated using $\{\rho^{r'}_i\}_{r' \in [r]}$ consistently with a subset of the incoming messages in $\{\mathcal{M}^{r',k \to i}_{in}\}_{r' < r, k \in [n]}$. Therefore, without loss of generality this can be cast as a locally consistent VRF-compliant attack.

Case 2: If $\tilde{m}^r_{i \to j} \notin \mathcal{M}^{r,i \to j}_{out}$, let $\{\tilde{\rho}^{r'}_i\}_{r' \in [r]}$ be the coins used by A' to generate $\tilde{m}^r_{i \to j}$. Then, $\tilde{\rho}^{r'}_i \neq \rho^{r'}_i$ for at least one r'. To provide a witness for the relation $\mathcal{R}^r_{i \to j}$, A' must generate $\tilde{\pi}^{r'}_i$ such that $\text{VRF.Verify}(vk^{vrf}_i, (i, r'), \tilde{\rho}^{r'}_i, \tilde{\pi}^{r'}_i) = 1$. By unique provability property of the VRF, such an attack can only succeed with negligible probability. □

The next claim considers stronger adversaries that are allowed to use arbitrary incoming messages for their next-message function.

Claim A.9. *Consider a perfect NIZK scheme. If Π_1 is a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent VRF-compliant adversaries, then Π' is a $(t, \alpha - \text{neg}(\kappa), \beta - \text{neg}(\kappa), q,$*

$\gamma - \mathrm{neg}(\kappa))$-$\mathsf{BA}$ *against locally consistent adversaries that are allowed to use arbitrary messages' contents when computing the next-message function.*

Proof. Consider an adversary A' that behaves locally consistent but can use arbitrary values as incoming messages. Assume that A' is VRF-compliant and let r be the first round in which A' deviates from the protocol with respect to incoming messages. Let P'_i be a corrupted party that uses $\{\tilde{\mathcal{M}}_{\mathsf{in}}^{r',k\to i}\}_{r'<r,k\in[n]}$ as its set of incoming messages to generate the message content for party P'_j in round r, denoted $\tilde{m}_{i\to j}^r$, and assume that $\bigcup \tilde{\mathcal{M}}_{\mathsf{in}}^{r',k\to i} \not\subseteq \bigcup \mathcal{M}_{\mathsf{in}}^{r',k\to i}$. There are two possible cases:

Case 1: If $\tilde{m}_{i\to j}^r \in \mathcal{M}_{\mathsf{out}}^{r,i\to j}$, then the adversary can compute a witness for the relation $\mathcal{R}_{i\to j}^r$. That is, even if $\bigcup \tilde{\mathcal{M}}_{\mathsf{in}}^{r',k\to i} \not\subseteq \bigcup \mathcal{M}_{\mathsf{in}}^{r',k\to i}$, the message $\tilde{m}_{i\to j}^r$ can be explained as if generated using a subset of $\bigcup \mathcal{M}_{\mathsf{in}}^{r',k\to i}$. Therefore, without loss of generality this can be cast as a locally consistent attack.

Case 2: If $\tilde{m}_{i\to j}^r \notin \mathcal{M}_{\mathsf{out}}^{r,i\to j}$, then to find a witness for the relation $\mathcal{R}_{i\to j}^r$, A' must produce for every message $\tilde{m}_{k\to i}^{r'} \in \bigcup \tilde{\mathcal{M}}_{\mathsf{in}}^{r',k\to i} \setminus \bigcup \mathcal{M}_{\mathsf{in}}^{r',k\to i}$ a signature $\tilde{\sigma}_{k\to i}^{r'}$, a VRF proof $\pi_{k\to i}^{r'}$ and a NIZK proof $\tilde{\varphi}_{k\to i}^{r'}$.

- If P_k is honest, A' can find an accepting signature $\tilde{\sigma}_{k\to i}^{r'}$ for $\tilde{m}_{k\to i}^{r'}$ under $\mathsf{vk}_k^{\mathsf{ds}}$ only with negligible probability (recall that every message $\tilde{m}_{k\to i}^{r'}$ encodes the values k,i,r'; hence, A' cannot reuse messages that were signed by P_k in other rounds).
- If P_k is corrupted, then in turn it must have provided a valid witness for the relation $\mathcal{R}_{k\to i}^{r'}$. By the minimality of r, it is guaranteed that $\tilde{m}_{k\to i}^{r'-1}$ was honestly generated with respect to the incoming messages of P'_k until round $r'-1$, $\{\bigcup \mathcal{M}_{\mathsf{in}}^{r'',k'\to k}\}_{r''\in[r'-1],k'\in[n]}$. In this case, without loss of generality, the message $\tilde{m}_{k\to i}^{r'}$ could have been sent by the corrupted P'_k to the corrupted P'_i, i.e., be included in the set $\mathcal{M}_{\mathsf{in}}^{r',k\to i}$.

The proof of the claim now reduces considering non-VRF-compliant adversaries, which follows from Claim A.8. $\qquad\square$

The next claim considers stronger adversaries that are not required to compute their outgoing messages by the next-message function, but can send arbitrary messages instead.

Claim A.10. *Consider a perfect NIZK scheme. If Π_1 is a $(t,\alpha,\beta,q,\gamma)$-$\mathsf{BA}$ against locally consistent VRF-compliant adversaries, then Π' is a $(t,\alpha - \mathrm{neg}(\kappa), \beta - \mathrm{neg}(\kappa), q, \gamma - \mathrm{neg}(\kappa))$-$\mathsf{BA}$ against malicious adversaries.*

Proof. Consider a malicious adversary A' and assume that A' behaves locally consistent and VRF-compliant until round r, i.e., round r is the first round in which A' does not compute a message according to the next-message function. Let P'_i be a corrupted party that generates the message content for party P'_j in round r, denoted $\tilde{m}_{i\to j}^r$, arbitrarily. There are two possible cases:

Case 1: If $\tilde{m}^r_{i \to j} \in \mathcal{M}^{r,i \to j}_{\text{out}}$, then the adversary can compute a witness for the relation $\mathcal{R}^r_{i \to j}$. That is, the message $\tilde{m}^r_{i \to j}$ can be explained as if generated using $\{\rho^{r'}_i\}_{r' \in [r]}$ consistently with a subset of the incoming messages in $\{\mathcal{M}^{r',k \to i}_{\text{in}}\}_{r' < r, k \in [n]}$ according to the next-message function. Therefore, without loss of generality this can be cast as a locally consistent VRF-compliant attack.

Case 2: If $\tilde{m}^r_{i \to j} \notin \mathcal{M}^{r,i \to j}_{\text{out}}$, then A' must provide $\tilde{\sigma}^r_{i \to j}$ and π^r_i along with a witness $\text{wit}^r_{i \to j}$ consisting of:

– An input bit x_i an setup_i.
– For every $r' \in [r]$ random coins $\rho^{r'}_i$.
– For every $r' \in [r-1]$ and $k \in [n]$ a message $\tilde{\boldsymbol{m}}^{r'}_{k \to i} = (\tilde{m}^{r'}_{k \to i}, \tilde{\sigma}^{r'}_{k \to i}, \pi^{r'}_k, \tilde{\varphi}^{r'}_{k \to i})$.

In addition it holds that $((\tilde{m}^r_{i \to j}, \tilde{\sigma}^r_{i \to j}, \pi^r_i), \text{wit}^r_{i \to j}) \in \mathcal{R}^r_{i \to j}$. As before, with all but negligible probability it is guaranteed that $\text{VRF.Verify}(\text{vk}^{\text{vrf}}_i, (i, r), \rho^r_i, \pi^r_i) = 1$ and for every honest party P'_k, $((\tilde{m}^{r'}_{k \to i}, \tilde{\sigma}^{r'}_{k \to i}, \pi^{r'}_k), \tilde{\varphi}^{r'}_{k \to i}) \in \mathcal{R}^{r'}_{k \to i}$. For a corrupted P'_k, if $((\tilde{m}^{r'}_{k \to i}, \tilde{\sigma}^{r'}_{k \to i}, \pi^{r'}_k), \tilde{\varphi}^{r'}_{k \to i}) \in \mathcal{R}^{r'}_{k \to i}$ then without loss of generality the message could have been sent by P'_k to P'_i. We conclude that with all but negligible probability, the $\tilde{m}^r_{i \to j}$ can be explained by a locally consistent VRF-compliant attack.

The proof of the claim now follows from Claim A.9. □

Finally, we remove the assumption of a perfect NIZK scheme and consider a NIZK scheme that allows for negligible adversarial advantage, and obtain the following claim.

Claim A.11. *If Π_1 is a $(t, \alpha, \beta, q, \gamma)$-BA against locally consistent VRF-compliant adversaries, then Π' is a $(t, \alpha - \text{neg}(\kappa), \beta - \text{neg}(\kappa), q, \gamma - \text{neg}(\kappa))$-BA against malicious adversaries.*

This concludes the proof of the first part of the theorem.

Proof of Item 2 **(public-randomness protocols).** We prove Item 2 of Theorem A.1 by adjusting the compiler Comp and removing the use of NIZK proofs. The new compiler Comp_{PR} is defined like Comp except that instead of computing a NIZK proof $\varphi^r_{i \to j} \leftarrow \text{NIZK.Prover}(\text{crs}, \text{stat}^r_{i \to j}, \text{wit}^r_{i \to j})$ for the relation $\mathcal{R}^r_{i \to j}$ and sending $\varphi^r_{i \to j}$, the sender P'_i simply sends the witness $\text{wit}^r_{i \to j}$. The receiver P'_j can now directly verify that $\text{wit}^r_{i \to j}$ is a valid witness. The proof follows immediately from Item 1 of Theorem A.1. □

References

[1] I. Abraham, T.H. Chan, D. Dolev, K. Nayak, R. Pass, L. Ren, E. Shi, Communication complexity of Byzantine agreement, revisited, in *Proceedings of the 38th Annual ACM Symposium on Principles of Distributed Computing (PODC)* (2019a), pp. 317–326

[2] I. Abraham, S. Devadas, D. Dolev, K. Nayak, L. Ren, Synchronous Byzantine agreement with expected $O(n)$ rounds, expected $o(n^2)$ communication, and optimal resilience, in *Financial Cryptography and Data Security* (2019b)

[3] H. Attiya, K. Censor, Tight bounds for asynchronous randomized consensus. *J. ACM*, 55(5):20:1–20:26 (2008)

[4] H. Attiya, K. Censor-Hillel, Lower bounds for randomized consensus under a weak adversary. *SIAM J. Comput.* 39(8):3885–3904 (2010)

[5] Z. Bar-Joseph, M. Ben-Or, A tight lower bound for randomized synchronous consensus, in *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 193–199 (1998)

[6] M. 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)*, pp. 27–30 (1983)

[7] M. Ben-Or, N. 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)*, pp. 408–416 (1985)

[8] M. Ben-Or, S. Goldwasser, A. 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)*, pp. 1–10 (1988)

[9] M. Ben-Or, E. Pavlov, V. 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)*, pp. 179–186 (2006)

[10] E. Ben-Sasson, A. Chiesa, M. Green, E. Tromer, M. Virza, Secure sampling of public parameters for succinct zero knowledge proofs, in *IEEE Symposium on Security and Privacy*, pp. 287–304 (2015)

[11] M. Blum, P. Feldman, S. Micali, Non-interactive zero-knowledge and its applications (extended abstract), in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 103–112 (1988)

[12] J. Bourgain, J. Kahn, G. Kalai, Influential coalitions for Boolean functions, in *CoRR*, 2014. arXiv:1409.3033

[13] S. Bowe, A. Gabizon, M.D. Green, A multi-party protocol for constructing the public parameters of the pinocchio zk-snark, in *Financial Cryptography and Data Security FC*, pp. 64–77 (2018)

[14] E. Boyle, R. Cohen, A. Goel, Breaking the o($\sqrt{}$ n)-bit barrier: Byzantine agreement with polylog bits per party, in *Proceedings of the 40th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 319–330 (2021)

[15] G. Bracha, An asynchronou [(n-1)/3]-resilient consensus protocol, in *Proceedings of the 3rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 154–162 (1984)

[16] M. Castro, B. Liskov. Practical Byzantine fault tolerance, in *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 173–186 (1999)

[17] D. Chaum, C. Crépeau, I. Damgård, Multiparty unconditionally secure protocols (extended abstract), in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 11–19 (1988)

[18] J. Chen, S. Micali, Algorand, in *CoRR*, 2016. arXiv:1607.01341

[19] B. Chor, B.A. Coan, A simple and efficient randomized Byzantine agreement algorithm, in *Fourth Symposium on Reliability in Distributed Software and Database Systems, SRDS*, pp. 98–106 (1984)

[20] B. Chor, M. Merritt, D.B. Shmoys, Simple constant-time consensus protocols in realistic failure models. *J. ACM*, 36(3):591–614 (1989)

[21] R. Cohen, S. Coretti, J.A. Garay, V. Zikas, Probabilistic termination and composability of cryptographic protocols, in *Advances in Cryptology – CRYPTO 2016, part III*, pp. 240–269 (2016)

[22] R. Cohen, S. Coretti, J. Garay, V. Zikas, Round-preserving parallel composition of probabilistic-termination cryptographic protocols, in *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, pp. 37:1–37:15 (2017)

[23] R. Cohen, I. Haitner, N. Makriyannis, M. Orland, A. Samorodnitsky, On the round complexity of randomized byzantine agreement, in *Proceedings of the 33st International Symposium on Distributed Computing (DISC)*, pp. 12:1–12:17 (2019)

[24] D. Dolev, R. Strong, Authenticated algorithms for Byzantine agreement. *SIAM J. Comput.* 12(4):656–666 (1983)

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

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

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

[28] M.J. Fischer, N.A. Lynch, M. Merritt, Easy impossibility proofs for distributed consensus problems, in *Proceedings of the 23th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 59–70 (1985)

[29] M. Fitzi, J.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)*, pp. 211–220 (2003)

[30] M. Fitzi, J.B. Nielsen, On the number of synchronous rounds sufficient for authenticated Byzantine agreement, in *Proceedings of the 23th International Symposium on Distributed Computing (DISC)*, pp. 449–463 (2009)

[31] E. Friedgut, Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica* 18(1):27–35 (1998)

[32] J.A. Garay, Y. Moses, Fully polynomial Byzantine agreement in t+1 rounds, in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 31–41 (1993)

[33] J.A. Garay, J. Katz, C. Koo, R. Ostrovsky, Round complexity of authenticated broadcast with a dishonest majority, in *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 658–668 (2007)

[34] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems, in *Advances in Cryptology – EUROCRYPT '99*, pp. 295–310 (1999)

[35] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, Algorand: Scaling Byzantine agreements for cryptocurrencies, in *Proceedings of the 26th Symposium on Operating Systems Principles (SOSP)*, pp. 51–68 (2017)

[36] O. Goldreich, S. Micali, A. 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)*, pp. 218–229 (1987)

[37] O. Goldreich, S. Goldwasser, N. Linial, Fault-tolerant computation in the full information model. *SIAM J. Comput.* 27(2):506–544 (1998)

[38] S. Goldwasser, S. Micali, R.L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.* 17(2):281–308 (1988)

[39] S. Goldwasser, E. Pavlov, V. Vaikuntanathan, Fault-tolerant distributed computing in full-information networks, in *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 15–26 (2006)

[40] S. Goldwasser, Y.T. Kalai, S. Park, Adaptively secure coin-flipping, revisited, in *Proceedings of the 42th International Colloquium on Automata, Languages, and Programming (ICALP), part II*, pp. 663–674 (2015)

[41] J. Groth, R. Ostrovsky, A. Sahai, New techniques for noninteractive zero-knowledge. *J. ACM* 59(3):11:1–11:35 (2012)

[42] V. Hadzilacos, Connectivity requirements for Byzantine agreement under restricted types of failures. *Distrib. Comput.* 2(2):95–103 (1987)

[43] D. Hofheinz, T. Jager, Verifiable random functions from standard assumptions, in *Proceedings of the 13th Theory of Cryptography Conference, TCC 2016-A, part I*, pp. 336–362 (2016)

[44] J. Kahn, G. Kalai, N. Linial, The influence of variables on Boolean functions (extended abstract), in *Proceedings of the 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 68–80 (1988)

[45] B.M. Kapron, D. Kempe, V. King, J. Saia, V. Sanwalani, Fast asynchronous Byzantine agreement and leader election with full information, in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 1038–1047 (2008)

[46] A.R. Karlin, A.C. Yao, Probabilistic lower bounds for Byzantine agreement and clock synchronization. Unpublished manuscript (1984)

[47] J. Katz, C. Koo, On expected constant-round protocols for Byzantine agreement, in *Advances in Cryptology – CRYPTO 2006*, pp. 445–462 (2006)

[48] V. King, J. Saia, Byzantine agreement in polynomial expected time: [extended abstract], in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 401–410 (2013)

[49] J. Kubiatowicz, D. Bindel, Y. Chen, S.E. Czerwinski, P.R. Eaton, D. Geels, R. Gummadi, S.C. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B.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*, pp. 190–201 (2000)

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

[51] A.B. Lewko, The contest between simplicity and efficiency in asynchronous Byzantine agreement, in *Proceedings of the 25th International Symposium on Distributed Computing (DISC)*, pp. 348–362 (2011)

[52] A.B. Lewko, M. Lewko, On the complexity of asynchronous agreement against powerful adversaries, in *Proceedings of the 32th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 280–289 (2013)

[53] Y. Lindell, A. Lysyanskaya, T. Rabin, On the composition of authenticated Byzantine agreement. *J. ACM*, 53(6):881–917 (2006)

[54] S. Micali, Very simple and efficient Byzantine agreement, in *Proceedings of the 8th Annual Innovations in Theoretical Computer Science (ITCS) conference*, pp. 6:1–6:1 (2017)

[55] S. Micali, V. Vaikuntanathan, Optimal and player-replaceable consensus with an honest majority. Unpublished manuscript (2017)

[56] S. Micali, M.O. Rabin, S.P. Vadhan, Verifiable random functions, in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 120–130 (1999)

[57] E. Mossel, R. O'Donnell, O. Regev, J. E. Steif, and B. Sudakov. Non-interactive correlation distillation, inhomogeneous Markov chains, and the reverse Bonami-Beckner inequality. *Israel Journal of Mathematics*, 154(1):299–336 (2006)

[58] E. Mossel, K. Oleszkiewicz, A. Sen, On reverse hypercontractivity. *Geom. Funct. Anal.* 23(3):1062–1097 (2013)

[59] G. Neiger, S. Toueg, Automatically increasing the fault-tolerance of distributed algorithms. *J. Algorithms* 11(3):374–419 (1990)

[60] R. O'Donnell, *Analysis of Boolean Functions* (Cambridge University Press, Cambridge, 2014)

[61] R. Pass and E. Shi. Hybrid consensus: Efficient consensus in the permissionless model, in *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, pp. 39:1–39:16 (2017)

[62] R. Pass, E. Shi, Thunderella: Blockchains with optimistic instant confirmation, in *Advances in Cryptology – EUROCRYPT 2018, part II*, pp. 3–33 (2018)

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

[64] T.P. Pedersen, Non-interactive and information-theoretic secure verifiable secret sharing, in *Advances in Cryptology – CRYPTO '91*, pp. 129–140 (1991)

[65] B. Pfitzmann, M. Waidner, Unconditional Byzantine agreement for any number of faulty processors, in *Proceedings of the 9th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pp. 339–350 (1992)

[66] M.O. Rabin, Randomized Byzantine generals, in *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 403–409 (1983)

[67] M. Santha, U.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)*, pp. 434–440 (1984)

[68] R. Turpin, B.A. Coan, Extending binary Byzantine agreement to multivalued Byzantine agreement. *Inf. Process. Lett.* 18(2):73–76 (1984)

[69] A.C. Yao, Protocols for secure computations (extended abstract), in *Proceedings of the 23th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 160–164 (1982)