Rational Broadcast Protocols against Timid Adversaries

Keigo Yamashita and Kenji Yasunaga

Tokyo Institute of Technology, Tokyo, Japan
keigo.elric.yamashita@gmail.com yasunaga@c.titech.ac.jp

Abstract. We present a constant-round deterministic broadcast protocol against *timid* adversaries in the synchronous authenticated setting. A timid adversary is a game-theoretically rational adversary who tries to attack the protocol but prefers the actions to be undetected. Our protocol is secure against such an adversary corrupting t out of n parties for any $t < n$. The round complexity is 5 for timid adversaries and is at most $t + 5$ for general malicious adversaries. Our results demonstrate that game-theoretic rationality enables us to circumvent the impossibility of constructing constant-round deterministic broadcast protocols for $t = \omega(1)$.

Keywords: Broadcast protocol · Game theory · Timid adversary

1 Introduction

Byzantine broadcast is a fundamental protocol in distributed computing used to construct fault-tolerant distributed systems and cryptographic protocols, including multiparty computation [28, 44, 43] and blockchains [27, 41, 35]. The Byzantine broadcast problem is that a specific party called the sender distributes a message among n parties in the presence of a malicious adversary who corrupts at most t parties. The difficulty is in a requirement that all non-corrupted parties should output the same value even if the sender is corrupted.

In synchronous networks with pairwise authenticated channels, the classical results [42, 40] show that broadcast is possible if and only if $t < n/3$. By assuming the setup of digital signatures, which is referred to as the *authenticated* setting, Dolev and Strong [17] presented a deterministic protocol with round complexity $t + 1$ for any $t < n$. They also showed the round complexity lower bound of $t + 1$ for deterministic protocols in the authenticated setting. Since then, many studies have been devoted to constructing randomized protocols with expected constant-round complexity [18, 24, 36, 1, 12, 48, 47].

In this work, we demonstrate that game-theoretic *rationality* can be used to circumvent the impossibility result of [17]. Specifically, we consider rational adversaries who prefer to violate the requirements of the broadcast protocol but do not prefer their actions to be detected. Namely, such adversaries prefer to attack the protocol stealthily. We call them *timid* adversaries.

Table 1. Previous and Our Results on Authenticated Broadcast Protocols

References	Resilience	Round Complexity	Adversary Model	Results
[17]	$t < n$	$t + 1$	Malicious	∃determ. protocol
[17]	$t < n$	t	Malicious	No determ. protocol
[20]	$t < n$	$t + 3$	Malicious	∃detectable protocol
[36]	$t < n/2$	57	Malicious	∃rand. protocol
[24]	$t < n/2 + k$	$O(k^2)$	Malicious	∃rand. protocol
[24]	$t < n$	$o(2n/(n-t))$	Malicious	No rand. protocol
[21]	$t < n/2 + k$	$O(k)$	Malicious	∃rand. protocol
[1]	$t < n/2$	10	Malicious	∃rand. protocol
[12]	$t < n$	$O(n/(n-t))$	Malicious	∃rand. protocol
This work	$t < n$	5	Rational	∃determ. protocol

A timid adversary is an adversary model that lies between semi-honest and malicious adversaries. A semi-honest adversary only tries to extract secret information by honestly performing the protocol. The model seems artificial and cannot be applied to protocols without secrecy requirements, such as broadcast. A malicious adversary does anything to attack the protocol and is a good model for studying the worst-case scenarios. However, the worst-case model restricts the usability of protocols and may not reflect real-life situations. A timid adversary attacks the protocol carefully so that his behavior will not be detected. Since the actions of a timid adversary vary depending on the detection mechanism of the protocol, we model it as a rational player in game theory who behaves to maximize his utility. A timid adversary can behave maliciously if the protocol does not employ any detection system. The adversary may behave like a semi-honest adversary if the protocol checks the validity of each computation.

Our Contributions. We introduce a game-theoretic security notion for broadcast protocols that takes into account adversaries' rational behavior. In our model, a single rational adversary corrupts a subset of participants of the broadcast protocol. The non-corrupted participants honestly follow the protocol. The adversary has a preference for the outcome of the protocol execution. We say a protocol is secure if (1) it satisfies the requirements for the broadcast protocol for a "harmless" adversary and (2) no timid adversary obtains higher utility than the harmless adversary. In other words, the protocol is secure in the sense that the best strategy for timid adversaries is doing nothing.

We construct a constant-round deterministic broadcast protocol against timid adversaries in the authenticated setting. The round complexity is 5 for timid adversaries and is at most $t+5$ for any malicious adversaries. The communication complexity of our protocol against timid adversaries is $O(\kappa n^2)$, where κ is a security parameter of the signature scheme. We summarize the previous and our results on authenticated broadcast protocols in Table 1.

The basic idea of our protocol is to use digital signatures as proofs/certificates. Consider a countersignature $\pi = (m, \sigma_B(\sigma_A(m)))$, where $\sigma_i(x)$ is a signature of player i for x. If player C has π, it means that C knows that player B has a proof that A has message m. Suppose that all players are prescribed to send the received countersignature to everyone by appending their own signature. If some player got the same countersignature as π from $t+1$ different players, it means that everyone knows that B has a proof that A has m. This is because there are at most t corrupted parties, and thus at least one honest party sent π to everyone. We use and generalize this idea to construct a constant-round protocol for timid adversaries.

Related Security Notions for Broadcast. We compare our security notion of rational broadcast against timid adversaries with the related notions in the literature.

In [19, 20], Fitzi et al. showed that *detectable broadcast* could be achieved for any $t < n$. In detectable broadcast, all honest parties either accept or reject the execution. A malicious adversary can cause an honest party to abort the protocol, but in that case, all the honest parties noticed the fact. Since a malicious action can be detected, any detectable broadcast protocol can be used as a rational protocol against timid adversaries. As far as we know, no constant-round detectable broadcast protocol exists.

Goldwasser and Lindell [29] presented a simple two-round protocol for *broadcast with abort* for any $t < n$. A requirement is relaxed such that any honest party needs to output either some same value or $\perp$. Since a broadcast protocol with abort may not have a mechanism for detecting malicious behaviors, the notion of broadcast with abort is incompatible with our security notion.

Aumann and Lindell [9] introduced the notion of *covert security*, where any deviation from the protocol can be detected with some probability ϵ. Although existing studies [30, 8, 39, 15] for covert security have been aimed at constructing general multiparty computation protocols, the security notion can be adopted to broadcast. If the probability ϵ is high enough, a protocol with covert security can be used as a broadcast protocol for timid adversaries. As observed in [45], the standard definition of covert security is not necessarily weaker than standard security against malicious adversaries. Since a secure Byzantine broadcast protocol is also secure for timid adversaries, the notion of covert security is strictly stronger than ours.

Our results of constructing a protocol that takes 5 rounds for rational adversaries and $t+5$ rounds for malicious adversaries are similar to the notion of *early stopping* [16], where the protocol may halt early if the actual number of corrupted parties is less than its maximum t. More specifically, Albouy et al. [5] studied the problem of constructing Byzantine broadcast protocols with good-case latency; they gave a deterministic broadcast protocol in the authenticated setting such that the round complexity may be a constant if the actual number of honest parties is sufficiently large. However, it is difficult to employ their protocol in our setting because it has no mechanism for detecting cheaters. Namely, a rational adversary may have no incentive to refrain from attacking the protocol.

Related Work. A game-theoretic analysis of players in cryptographic protocols was initiated by Halpern and Teague [33] for secret sharing. The problem is achieving fair secret reconstruction among rational parties, which has been extensively studied in the literature. See [2, 38, 22, 7, 37] and the references therein. Fairness among rational parties has been studied for other problems such as multiparty computation [6, 31], leader election [3, 4, 49, 13], consensus [34], and coin toss [14].

There have been studies on protocols against rational adversaries to circumvent the known impossibility results. Groce et al. [32] studied the possibility of constructing a Byzantine agreement tolerating t corruptions for $t \geq n/2$, which is impossible in the traditional setting. Garay et al. [25] introduced a framework of rational protocol design to capture incentive-driven adversaries within the simulation-based paradigm. Their framework was used to relax fairness in multiparty computation [26] and analyze Bitcoin [10]. The notion of timid adversaries was introduced by Fujita et al. [23] as a game-theoretically relaxed model of a malicious adversary. They presented perfectly secure message transmission protocols that circumvent the known impossibility results.

2 Preliminaries

We briefly describe our network model, the setup assumptions, and the definition of Byzantine broadcast.

There are n parties on the network. A protocol is said to be *t-resilient* if it works correctly, even if at most t parties are corrupted and controlled by an adversary. We assume the synchronous communication model. Namely, the protocol proceeds in rounds, and each party can send messages to other parties in each round. The messages of non-corrupted (honest) parties can be correctly delivered at the beginning of the next round.

We assume a public-key infrastructure (PKI) and digital signature schemes. Each party can generate a signature using his secret key, and the validity can be checked with the corresponding public key. It is called an *authenticated* setting.

A signature scheme consists of three algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$. A key-generation algorithm Gen, on input security parameter n, outputs a pair of keys (pk, sk). The security parameter is usually represented by the string $1 \cdots 1$ of length n, denoted by 1^n. A signing algorithm Sign, on input secret key sk and message m, outputs a signature σ. A verification algorithm Ver, on input public key pk and pair (m, σ), checks if σ is a valid signature of m. Here, we give a formal definition of the standard security notion of signature schemes.

Definition 1 (Security of Signature Scheme). *A signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is* existentially unforgeable against chosen-message attack (EUF-CMA) *or simply* secure *if for every polynomial-time adversary A,*

$$\Pr\left[(pk, sk) \leftarrow \mathsf{Gen}(1^n); (m, \sigma) \leftarrow A^{\mathsf{Sign}_{sk}(\cdot)}(vk) : m \notin Q \wedge \mathsf{Ver}_{pk}(m, \sigma) = 1\right]$$

is negligible in n, where $Q = \{m_i\}_i$ is the set of queries m_i made by A to oracle $\mathsf{Sign}_{sk}(\cdot)$, for which A received σ_i generated by $\mathsf{Sign}_{sk}(m_i)$ as a response.

In the above definition, an adversary A can use the signing oracle $\mathsf{Sign}_{sk}(\cdot)$ as many times as A wants to obtain valid pairs $\{(m_i, \sigma_i)\}_i$ of message m_i and signature σ_i, where each m_i was chosen by A. Finally, A outputs a pair (m, σ). The winning condition that $m \notin Q \wedge \mathsf{Ver}_{pk}(m, \sigma) = 1$ means that the submitted message m should differ from the messages queried to the signing oracle, and the pair should be a valid message-signature pair. Thus, the above security guarantees that no adversary can generate a valid signature-message pair except those generated by a valid signing algorithm.

As a correctness property, we require that for any (pk, sk) generated by $\mathsf{Gen}(1^n)$ and message m, it holds that $\mathsf{Ver}_{vk}(m, \mathsf{Sign}_{sk}(m)) = 1$. For simplicity, we assume an ideal signature scheme where the above probability is equal to zero.

The following is a traditional definition of Byzantine broadcast.

Definition 2 (Byzantine Broadcast). *A protocol Π for n parties is said to be a* t-resilient Byzantine broadcast protocol *if the following conditions hold for any adversary controlling at most t parties:*

1. *Validity: If the sender is honest and holds an initial input m, then all honest parties output m.*
2. *Agreement: All honest parties output the same value.*

Dolev and Strong [17] presented a polynomial-time authenticated broadcast protocol with round complexity $t + 1$ for any $t < n$. Also, they showed that as long as protocols are deterministic, the round complexity must be at least $t + 1$, even in the authenticated setting.

3 Rational Broadcast Protocols

We define a game-theoretically rational adversary model. First, we define a game played by a rational player/adversary. The outcome of the game consists of the information that represents whether the adversary successfully violates the security requirements. Since timid adversaries care whether their actions were detected, the outcome also includes such information. After that, we define a security notion of rational broadcast protocols, which roughly says that the best strategy for rational adversaries is doing nothing on the protocol.

Broadcast Game. We define the *broadcast game*. First, set parameters $\mathsf{incorrect} = \mathsf{disagree} = \mathsf{undetect} = 0$. Given the protocol Π, an adversary A chooses the sender $s \in [n]$, the message m, and the set of parties $C \subseteq [n]$ with $|C| \leq t$. The protocol is executed by specifying s as the sender with initial message m, where the parties in C are controlled by A, and the other parties honestly follow the protocol description of Π. After running the protocol, each party $i \in [n]$ outputs val_i. Let $H = [n] \setminus C$. If $s \in H$ and there exists $i \in H$ such that $\mathsf{val}_i \notin \{m, \perp\}$, set $\mathsf{incorrect} = 1$. If there exist $i, j \in H$ such that $\mathsf{val}_i \neq \mathsf{val}_j$, set $\mathsf{disagree} = 1$. In executing the protocol, every player may send a message "DETECT i"

indicating that the player detected that player i cheating. If no player sent messages "DETECT i" during the protocol for $i \in C$, set $\mathsf{undetect} = 1$. The outcome of the game against adversary A is $\mathsf{out}_A = (\mathsf{incorrect}, \mathsf{disagree}, \mathsf{undetect})$.

In the above definition of $\mathsf{incorrect}$, the case that $\mathsf{val}_i = \perp$ for $i \in H$ is not considered a successful attack by adversaries. One reason is that if $\mathsf{val}_i = \perp$ for some honest party i, i may propose to execute the protocol again. If so, we cannot say that the adversary attacked successfully. Another reason is that the output value $\perp$ usually implies that some attack was detected. Hence, timid adversaries naturally consider that outcome a failure.

Utility. The utility $u(A)$ of the adversary A is the expected value $\mathbb{E}[U(\mathsf{out}_A)]$, where U is a function that maps the outcome out_A of the game to real values.

Definition 3 (Security of Rational Broadcast). *A broadcast protocol Π is said to be* secure against rational t-adversaries with utility function U *if there exists a "harmless" adversary B controlling at most t parties such that*

1. *Security: Π satisfies validity and agreement for B;*
2. *Nash equilibrium: For any adversary A controlling at most t parties, $u(A) \leq u(B)$.*

The above notion captures game-theoretic security; if protocol Π satisfies the above, a strategy of harmless adversary B is the best response since every other strategy (following adversary A) cannot increase the expected utility. Thus, every adversary rationally behaves harmlessly in protocol Π.

Timid Adversaries. We consider a *timid* adversary who tries to violate the security requirements of protocols but does not prefer the attacks to be detected. Specifically, we consider the set of utility functions that satisfy the following conditions:

1. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{incorrect} > \mathsf{incorrect}'$, $\mathsf{disagree} = \mathsf{disagree}'$, and $\mathsf{undetect} = \mathsf{undetect}'$;
2. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{incorrect} = \mathsf{incorrect}'$, $\mathsf{disagree} > \mathsf{disagree}'$, and $\mathsf{undetect} = \mathsf{undetect}'$;
3. $U(\mathsf{out}) > U(\mathsf{out}')$ if $\mathsf{incorrect} = \mathsf{incorrect}'$, $\mathsf{disagree} = \mathsf{disagree}'$, and $\mathsf{undetect} > \mathsf{undetect}'$,

where $\mathsf{out} = (\mathsf{incorrect}, \mathsf{disagree}, \mathsf{undetect})$ and $\mathsf{out}' = (\mathsf{incorrect}', \mathsf{disagree}', \mathsf{undetect}')$ are two outcomes of the broadcast game. We denote by U_{timid} the set of utility functions satisfying the above conditions.

By definition, for any $U \in U_{\mathsf{timid}}$, it holds that

$$\begin{aligned} U(1,1,1) &> \max\{U(0,1,1), U(1,0,1)\} \\ &\geq \min\{U(0,1,1), U(1,0,1)\} > U(0,0,1) > U(0,0,0). \end{aligned}$$

We use the above relation in the analysis.

Note that if protocol Π satisfies t-resilient Byzantine broadcast of Definition 2, any adversary controlling at most t parties achieves either $U(0,0,0)$ or $U(0,0,1)$. Since a harmless adversary will achieve $U(0,0,1)$, Π is also secure against rational t-adversaries. Namely, Definition 3 for timid adversaries is a relaxation of Definition 2.

4 Our Protocol

We assume that a PKI is established on the network. Let $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. We assume that each party $i \in [n]$ has a pair (pk_i, sk_i) of keys generated by $\mathsf{Gen}(1^n)$ and all parties know $\{pk_i\}_{i\in[n]}$. With the secret key sk_i, party i can generate a signature $\sigma_i(m)$ of message m by $\mathsf{Sign}_{sk_i}(m)$. The validity of a pair (m, σ_i) can be verified with the public key pk_i by $\mathsf{Ver}_{pk_i}(m, \sigma_i)$.

First, we recall the Dolev-Strong authenticated broadcast protocol [17]. The protocol uses a *signature chain*. A signature chain for value v of length ℓ is defined as (1) $(v, \sigma_i(v))$ for some $i \in [n]$ if $\ell = 1$; (2) $(c, \sigma_i(c))$ for some $i \in [n]$ for $\ell > 1$, where c is a signature chain for v of length $\ell - 1$ that consists of signatures with $\ell - 1$ distinct signers other than i. A signature chain is valid if it satisfies the above conditions and all signatures are valid.

Dolev-Strong Protocol.

1. The sender s with input m sends $(m, \sigma_s(m))$ to all parties.
2. For round $r = 2, \ldots, t+1$, each party i does the following:
   - If i received a valid signature chain c for value v of length $r-1$ where no signature of i is included, then i signs it and sends $(c, \sigma_i(c))$ to all parties. (Party i does this procedure once for each value v. Namely, if i appended a signature for value v and sent to all parties, i does nothing for value v henceforth.)
   - At the end of round $t+1$, let V be the set of values of valid signature chains of length $t+1$ that i received. If $|V| = 0$ or $|V| > 1$, i outputs $\perp$. Otherwise, i output the value in V.

Before presenting the formal description, we give an overview of our protocol. In the following, we introduce three notions: proof of dissemination (PoD), proof of agreement (PoA), and proof of termination (PoT). They help us understand our protocol and make the security proof easy to follow.

Protocol Overview.

1. The sender sends the initial input m and its signature to all parties.
2. Each party generates a countersignature from the received message and sends it to all parties.
3. Each party collects countersignatures. A set of $t+1$ valid countersignatures functions as a "proof of dissemination" of message m. It means that a non-corrupted party has sent a countersignature of m to all parties. Party i sends the local proof PoD^i_m for message m to all parties. If i found valid countersignatures for different values, i does nothing.

$$\mathsf{PoD}^i_m = \text{“Party } i \text{ knows that everyone got the proof that } s \text{ sent } m.\text{”}$$

Note that, even if party i has PoD^i_m, there may be the case that s sent $m' \neq m$ to some party.

4. Each party collects proofs of dissemination $\{\mathsf{PoD}^j_m\}$. A set of $t+1$ valid proofs for consistent m is a “proof of agreement,” implying that a non-corrupted party has found no inconsistency and sent a proof of dissemination to all parties. If party i gets a proof of agreement $\mathsf{PoA}^i_m = \{\mathsf{PoD}^j_m\}_j$, i sends the local proof PoA^i_m to all parties via the Dolev-Strong protocol. Otherwise, party i does nothing.

$$\mathsf{PoA}^i_m = \begin{array}{c}\text{“Party } i \text{ knows that everyone knows that}\\ \text{everyone got the proof that } s \text{ sent } m.\text{”}\end{array}$$

Even if party i has PoA^i_m, there may be the case that another party j does not have PoA^j_m. Namely, j got the proof that s sent m, but j does not know everyone knows this fact.

5. A set of $t+1$ valid proofs $\{\mathsf{PoA}^j_m\}$ works as a “proof of termination” since it implies that a valid proof of agreement has been sent to all parties. If party i gets a proof of termination $\mathsf{PoT} = \{\mathsf{PoA}^j_m\}_j$, i outputs the value m. If another party j has not obtained a proof of termination, j continues to run the Dolev-Strong protocol, in which party i also needs to participate. At the end of the Dolev-Strong protocol, if party i found a valid PoA^j_m, i outputs m. Otherwise, i outputs $\perp$ and sends a message “DETECT s,” meaning that the sender s has cheated.

$$\mathsf{PoT} = \begin{array}{c}\text{“Everyone knows that everyone knows that}\\ \text{everyone got the proof that } s \text{ sent } m.\text{”}\end{array}$$

We give a formal description of our protocol. Since we define several *validity* notions, we summarize them in Table 2.

Our Protocol Π_{rbc}.

Note that, in each round, if party i received a message containing $(x, \sigma_j(x))$ from party j such that $\sigma_j(x)$ is not a valid signature, then i considers j has sent i nothing.

1. The sender s with input m sends $(m, \sigma_s(m))$ to all parties.
2. For each party i, if i received a valid signature $(m, \sigma_s(m))$ from s and received no valid signature for other value $m' \neq m$, then i signs it and sends the countersignature $(m, \sigma_i(\sigma_s(m)))$ to all parties. Otherwise, i sends nothing.
3. For each party i, if i received at least $t+1$ valid countersignatures of distinct signers for the same value m and did not see any valid countersignature for other value $m' \neq m$, then i sends a proof of dissemination

$$\mathsf{PoD}^i_m = (m, \mathsf{CSigSet}^i_m, \sigma_i(\mathsf{CSigSet}^i_m))$$

Table 2. Validity Notions

Objects	Validity Conditions
Signature $\sigma_i(m)$	$\mathsf{Ver}_{vk_i}(m, \sigma_i(m)) = 1$
Countersignature $\sigma_i(\sigma_s(m))$	$\mathsf{Ver}_{vk_i}(\sigma_s(m), \sigma_i(\sigma_s(m))) = 1$ $\wedge\ \mathsf{Ver}_{vk_s}(m, \sigma_s(m)) = 1$
Countersignature set $\mathsf{CSigSet}^i_m = \{\sigma_j(\sigma_s(m))\}_j$	$\forall j$, $\sigma_j(\sigma_s(m))$ is valid $\wedge$ each j in $\mathsf{CSigSet}^i_m$ is distinct $\wedge\ \lvert\mathsf{CSigSet}^i_m\rvert \geq t+1$
Proof of dissemination $\mathsf{PoD}^i_m = (m, \mathsf{CSigSet}^i_m, \sigma^i_m)$	$\mathsf{Ver}_{vk_i}((m, \mathsf{CSigSet}^i_m), \sigma^i_m) = 1$ $\wedge\ \ \mathsf{CSigSet}^i_m$ is valid
(Signed) proof of agreement $\mathsf{PoA}^i_m = (\mathsf{PoA}_m, \sigma^i_m)$, where $\mathsf{PoA}_m = (m, \{(\mathsf{CSigSet}^j_m, \sigma^j_m)\}_j)$	$\mathsf{Ver}_{sk_i}(\mathsf{PoA}_m, \sigma^i_m) = 1$ $\wedge\ \forall j, \big(\mathsf{Ver}_{vk_j}(\mathsf{CSigSet}^j_m, \sigma^j_m) = 1$ $\wedge\ \mathsf{CSigSet}^j_m$ is valid $\big)$ $\wedge$ each j of $\mathsf{CSigSet}^j_m$ is distinct $\wedge\ \lvert\{(\mathsf{CSigSet}^j_m, \sigma^j_m)\}_j\rvert \geq t+1$
Signature chain $C^j = (\mathsf{PoA}^j_m, \sigma^j_m)$ of length k	$\sigma^j_m = \sigma_{i_k}(\sigma_{i_{k-1}}(\cdots(\sigma_{i_1}(\mathsf{PoA}^j_m))\cdots))$ $\wedge\ \mathsf{PoA}^j_m$ is valid $\wedge\ \forall \ell \in [k], \sigma_{i_\ell}(\cdots)$ is valid $\wedge$ each i_ℓ in σ^j_m is distinct $\wedge$ received in round $4+k$

to all parties, where

$$\mathsf{CSigSet}^i_m = \{\sigma_j(\sigma_s(m))\}_j$$

is the set of valid countersignatures of distinct signers for m that i received and $|\mathsf{CSigSet}^i_m| \geq t+1$.
Otherwise, i sends nothing.

4. For each party i, if i received at least $t+1$ valid proofs of dissemination $\{\mathsf{PoD}^j_m\}$ of distinct j for the same value m and did not see any valid proof for other value $m' \neq m$, then i sends a signed proof of agreement $\mathsf{PoA}^i_m = (\mathsf{PoA}_m, \sigma_i(\mathsf{PoA}_m))$ to all parties, where

$$\mathsf{PoA}_m = (m, \{(\mathsf{CSigSet}^j_m, \sigma_j(\mathsf{CSigSet}^j_m))\}_j)$$

is generated from a set of valid proofs of dissemination of distinct j for m that i received and $|\{(\mathsf{CSigSet}^j_m, \sigma^j_m)\}_j| \geq t+1$.
Otherwise, i sends nothing.

5. For round $r = 4+k$ with $k = 1, \ldots, t+1$, each party i does the following:
   (a) In each round, if i received from j a valid signature chain $C^j = (\mathsf{PoA}^j_m, \sigma^j_m)$ containing no signature of i and did not see any valid chain for other value $m' \neq m$, i sends $(\mathsf{PoA}^j_m, \sigma_i(\sigma^j_m))$ to all parties. (Note that i does this procedure once for each value PoA^j_m.)
   If i obtained at least $t+1$ signed proofs of agreement $\{(\mathsf{PoA}^j_m, \sigma_\ell(\sigma^j_m))\}_\ell$ (including i's one) with valid signatures of distinct ℓ for the same value

m and did not see any valid proof for other value $m' \neq m$, i outputs m and halts.
Otherwise, i sends nothing.

(b) At the end of round $t+5$, if party i received a valid signature chain of length $t+1$ containing valid PoA^j_m and did not see any valid proof for other value $m' \neq m$, i outputs m and halts.
Otherwise, i sends "DETECT s" to all parties, outputs $\perp$, and halts.

4.1 Security Proofs

We give a security proof of our protocol. Before proving the main theorem (Theorem 1), we give a technical lemma used in the proof.

Lemma 1. *In every broadcast game of Π_{rbc} in the presence of rational t-adversary with utility function $U \in U_{\mathsf{timid}}$ for $t < n$, it holds that (1) if $i \in H$ outputs $m \neq \perp$, then i have obtained a valid PoA^j_m for some $j \in [n]$; (2) if $i \in H$ outputs $\perp$, every $\ell \in H$ have failed to generate a signed proof of agreement $(\mathsf{PoA}^j_m, \sigma_\ell(\mathsf{PoA}^j_m))$ for a valid PoA^j_m for some $j \in [n]$ in round 4.*

Proof. Since every $i \in H$ follows the prescribed protocol, we can see that i outputs $m \neq \perp$ in round 5 or $t+5$. For the former case, i obtained at least $t+1$ signed proofs of agreement $\{(\mathsf{PoA}^j_m, \sigma_\ell(\sigma^j_m))\}_\ell$; for the latter case, i received a valid signature chain containing valid PoA^j_m. Thus, in both cases, $i \in H$ have obtained a valid PoA^j_m, implying (1).

Similarly, $i \in H$ outputs $\perp$ only when i failed to obtain a valid signature chain of length $t+1$ in round $t+5$. This event happens only when every honest party ℓ failed to obtain a valid PoA^j_m in round 4; this is because if $\ell \in H$ obtained a valid PoA^j_m, ℓ performs the Dolev-Strong protocol as a sender to broadcast $(\mathsf{PoA}^j_m, \sigma_\ell(\mathsf{PoA}^j_m))$ to all parties. By the agreement property of the Dolev-Strong protocol, honest party i would obtain a valid PoA^j_m, a contradiction. Hence, (2) follows. □

Theorem 1. *The broadcast protocol Π_{rbc} is secure against rational t-adversaries with utility function $U \in U_{\mathsf{timid}}$ for any $t < n$. The round complexity is 5 for a harmless adversary and is at most $t+5$ for any adversary controlling t parties.*

Proof. We consider a harmless adversary B that chooses a random sender $s \in [n]$, a random message m, and $C = \emptyset$. Namely, B does not make any attacks on the protocol. It is not difficult to see that the protocol satisfies validity and agreement against B. In the presence of B, each party i receives n valid proofs of agreement in round 5. Thus, the round complexity for a harmless adversary is 5.

We show the Nash equilibrium property. Since $u(B) = U(0,0,1)$, we need to show that for any adversary A, $u(A) \leq U(0,0,1)$. To achieve a higher utility, an adversary needs outcomes of $\mathsf{incorrect} = 1$ or $\mathsf{disagree} = 1$.

Consider the case that $\mathsf{incorrect} = 1$. By definition, when $\mathsf{incorrect} = 1$, the sender must not be corrupted. Since no party other than s can generate $\sigma_s(m')$

for $m' \neq m$, parties will not output messages other than m or $\perp$. Namely, as long as the signature scheme is unforgeable, it is not possible to be the case that $\mathsf{incorrect} = 1$.

Next, consider the case that $\mathsf{disagree} = 1$. Suppose for contradiction that two parties $i, j \in H$ output $\mathsf{val}_i = m, \mathsf{val}_j = m' \neq m$, respectively.

First, we consider the case that $\perp \notin \{m, m'\}$. By (1) of Lemma 1, i and j have obtained valid PoA_m and $\mathsf{PoA}_{m'}$, respectively. A valid PoA_m contains a set $\{\mathsf{CSigSet}_m^\ell\}_\ell$ of size at least $t+1$ for distinct ℓ, where each $\mathsf{CSigSet}_m^\ell$ is valid. Since there are at most t corrupted parties, the existence of valid PoA_m implies that some non-corrupted party ℓ sent $\mathsf{CSigSet}_m^\ell$ to all parties in round 3. Since each $\mathsf{CSigSet}_m^\ell$ consists of at least $t+1$ valid countersignatures for m, some non-corrupted party ℓ' sent $(m, \sigma_{\ell'}(\sigma_s(m)))$ to all parties in round 2. Similarly, one can deduce that the existence of valid $\mathsf{PoA}_{m'}$ implies that some party ℓ'' sent $(m', \sigma_{\ell''}(\sigma_s(m')))$ to all parties in round 2. Thus, all parties must have received valid countersignatures for distinct m and m'. In that case, all non-corrupted parties would have sent nothing in round 3, a contradiction.

Next, we consider the case that $m \neq \perp$ and $m' = \perp$. Since $\mathsf{val}_j = \perp$, (2) of Lemma 1 implies that every $\ell \in H$ has failed to generate a signed proof of agreement. In that case, since there are at most t corrupted parties, no party can receive at least $t+1$ valid proofs of agreement $\{\mathsf{PoA}^j\}$ of distinct j in round 5. Thus, it must be the case that party i output m after performing the Dolev-Strong protocol. By the agreement property of the Dolev-Strong protocol, party $j \in H$ would output m, contradicting the fact that $\mathsf{val}_j = \perp$. Thus, it is impossible to achieve $\mathsf{disagree} = 1$.

By the above analysis, for any adversary A, the utility $u(A)$ is either $U(0,0,1)$ or $U(0,0,0)$. Note that $u(A) = U(0,0,0)$ when A corrupts the sender s, the protocol halts in round $t+5$, and the cheating of s is detected. Since $u(A) \leq U(0,0,1) = u(B)$, the protocol satisfies a Nash equilibrium.

To prove the worst-case round complexity, consider the case that some party i sent a valid signature chain $C^i = (\mathsf{PoA}_m^i, \sigma_m^i)$ of length k to some honest party j in round $4+k$ for some $k = 1, \ldots, t+1$. In that case, by the property of the Dolev-Strong protocol, every honest party can obtain a valid signature chain of length $t+1$ in round $t+5$. Otherwise, no honest party will receive a valid signature chain of length $t+1$ in round $t+5$, and thus all honest parties output $\perp$ by sending "DETECT s". In either case, the worst-case round complexity is $t+5$. □

Communication complexity. The communication complexity of the above protocol against a harmless adversary is $O(\kappa n^3)$, where κ is a security parameter of the signature scheme, and we assume that each signature is of length $O(\kappa)$. We can employ *non-interactive threshold signatures* [46, 11] to reduce the communication complexity. In a non-interactive threshold signature scheme, each party can generate a signature share of message x, and there is an algorithm that converts k valid shares to a signature of x. No set of less than k parties can forge a valid signature. In our protocol, a set $\mathsf{CSigSet}_m^j$ of countersignatures can be replaced with a threshold signature. Namely, in round 2, each party sends a

signature share of $(m, \sigma_s(m))$ to all parties, and in round 3, each party generates a valid threshold signature of $(m, \sigma_s(m))$ instead of $\mathsf{CSigSet}^i_m$. Since the size of PoD^i_m can be reduced from $O(\kappa n)$ to $O(\kappa)$, the total communication complexity of the resulting protocol is $O(\kappa n^2)$.

4.2 Detecting Cheaters

In our protocol, the sender is the only player who can be detected as a cheater. Regarding this point, we can show that as long as $t \leq \lfloor (n-1)/2 \rfloor$, the sender s can be declared a cheater only when s is corrupted.

Proposition 1. *In every broadcast game of Π_{rbc} in the presence of rational t-adversary A with utility function $U \in U_{\mathsf{timid}}$ for $t \leq \lfloor (n-1)/2 \rfloor$, if $i \in H$ outputs $\perp$, then A chose the sender $s \in [n]$ and $C \subseteq [n]$ such that $s \in C$.*

Proof. Suppose for contradiction that A chose the sender $s \in [n]$ and C such that $s \notin C$. Since $s \in H$, every player receives a valid signature $\sigma_s(m)$ in round 2. Then, every player $i \in H$ sends a valid countersignature $\sigma_i(\sigma_s(m))$ to all players. Since the number of honest players satisfies

$$|H| = n - t \geq n - \left\lfloor \frac{n-1}{2} \right\rfloor \geq \left\lfloor \frac{2n-(n-1)}{2} \right\rfloor = \left\lfloor \frac{n-1}{2} + 1 \right\rfloor \geq t + 1,$$

every player $i \in H$ obtains at least $t+1$ valid countersignatures in round 3. By a similar argument, every player $i \in H$ obtains at least $t+1$ valid proofs of dissemination in round 4 and obtains at least $t+1$ valid signed proofs of agreement in round 5. Hence, every player $i \in H$ outputs m in round 5 and halts, which contradicts the assumption that some honest player outputs $\perp$. Therefore, the statement follows. □

Proposition 1 guarantees a sort of soundness of the detection mechanism in our protocol. However, we can see that the guaranteed bound $t \leq \lfloor (n-1)/2 \rfloor$ is optimal and cannot be extended to $t > \lfloor (n-1)/2 \rfloor$. Specifically, there is a t-adversary A with $t = \lfloor (n-1)/2 \rfloor + 1$ such that A chooses the sender s and C with $s \notin C$, but every $i \in H$ outputs $\perp$. The strategy of A is fairly simple; every party $i \in C$ does nothing in the protocol. For such A, the number of honest players satisfies

$$|H| = n - t = n - \left(\left\lfloor \frac{n-1}{2} \right\rfloor + 1 \right) \leq t.$$

Since only at most t players are active, every honest player cannot generate any valid countersignature set, for which at least $t+1$ valid countersignatures are needed. Thus, honest players will output $\perp$ in the game against A.

The above weakness of our protocol does not contradict the game-theoretic security of Definition 3. Since every honest player outputs $\perp$, the outcome of the game is $\mathsf{out}_A = (\mathsf{incorrect}, \mathsf{disagree}, \mathsf{undetect}) = (0, 0, 1)$, where $s \in H$ is wrongly detected as a cheater, but no player $i \in C$ is detected. The above strategy of A achieves the same utility as a harmless one and does not violate a Nash equilibrium.

5 Discussion

In this work, we introduce a game-theoretic security notion for broadcast protocols, which can be used in various cryptographic protocols such as multiparty computation and blockchains. We have developed a constant-round broadcast protocol against adversaries corrupting t out of n players for any $t < n$. Since constructing constant-round protocols is impossible for malicious adversaries, our protocol heavily relies on the rationality of timid adversaries who prefer their actions to be undetected in protocol executions.

There are several interesting open problems. First, as discussed in Section 4.2, our protocol may wrongly detect an honest player as a cheater for $t \geq \lfloor (n-1)/2 \rfloor + 1$. Possible future work is constructing a protocol without such weakness or proving it is impossible. Another one is improving our protocol with respect to round complexity and communication complexity. The worst-case round complexity of our protocol is $t + 5$, which depends on the number of corrupted players. It may be interesting to incorporate randomized protocols ([1, 12, 48, 47]) instead of the Dolev-Strong protocol [17] for constructing protocols with expected constant-round protocols for (worst-case) malicious adversaries.

Acknowledgments This study was supported in part by JSPS KAKENHI Grant Numbers 23H00468 and 23K17455. The second author would like to thank Toshihiko Ishihara for the discussion at an early stage of this work.

References

1. I. Abraham, S. Devadas, D. Dolev, K. Nayak, and L. Ren. Synchronous byzantine agreement with expected O(1) rounds, expected o(n^2) communication, and optimal resilience. In I. Goldberg and T. Moore, editors, *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers*, volume 11598 of *Lecture Notes in Computer Science*, pages 320–334. Springer, 2019.
2. I. Abraham, D. Dolev, R. Gonen, and J. Y. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In E. Ruppert and D. Malkhi, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*, pages 53–62. ACM, 2006.
3. I. Abraham, D. Dolev, and J. Y. Halpern. Distributed protocols for leader election: A game-theoretic perspective. *ACM Trans. Economics and Comput.*, 7(1):4:1–4:26, 2019.
4. Y. Afek, Y. Ginzberg, S. L. Feibish, and M. Sulamy. Distributed computing building blocks for rational agents. In M. M. Halldórsson and S. Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 406–415. ACM, 2014.
5. T. Albouy, D. Frey, M. Raynal, and F. Taïani. Good-case early-stopping latency of synchronous byzantine reliable broadcast: The deterministic case. In C. Scheideler, editor, *36th International Symposium on Distributed Computing, DISC 2022, October 25-27, 2022, Augusta, Georgia, USA*, volume 246 of *LIPIcs*, pages 4:1–4:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

6. G. Asharov, R. Canetti, and C. Hazay. Toward a game theoretic view of secure computation. *J. Cryptol.*, 29(4):879–926, 2016.
7. G. Asharov and Y. Lindell. Utility dependence in correct and fair rational secret sharing. *J. Cryptol.*, 24(1):157–202, 2011.
8. G. Asharov and C. Orlandi. Calling out cheaters: Covert security with public verifiability. In X. Wang and K. Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 681–698. Springer, 2012.
9. Y. Aumann and Y. Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. *J. Cryptol.*, 23(2):281–343, 2010.
10. C. Badertscher, J. A. Garay, U. Maurer, D. Tschudi, and V. Zikas. But why does it work? A rational protocol design treatment of bitcoin. In J. B. Nielsen and V. Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 34–65. Springer, 2018.
11. C. Cachin, K. Kursawe, and V. Shoup. Random oracles in constantipole: practical asynchronous byzantine agreement using cryptography (extended abstract). In G. Neiger, editor, *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing, July 16-19, 2000, Portland, Oregon, USA*, pages 123–132. ACM, 2000.
12. T. H. Chan, R. Pass, and E. Shi. Sublinear-round byzantine agreement under corrupt majority. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 246–265. Springer, 2020.
13. K. Chung, T. H. Chan, T. Wen, and E. Shi. Game-theoretic fairness meets multiparty protocols: The case of leader election. In T. Malkin and C. Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2021.
14. K. Chung, Y. Guo, W. Lin, R. Pass, and E. Shi. Game theoretic notions of fairness in multi-party coin toss. In A. Beimel and S. Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part I*, volume 11239 of *Lecture Notes in Computer Science*, pages 563–596. Springer, 2018.
15. I. Damgård, C. Orlandi, and M. Simkin. Black-box transformations from passive to covert security with public verifiability. In D. Micciancio and T. Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 647–676. Springer, 2020.
16. D. Dolev, R. Reischuk, and H. R. Strong. Early stopping in byzantine agreement. *J. ACM*, 37(4):720–741, 1990.
17. D. Dolev and H. R. Strong. Authenticated algorithms for byzantine agreement. *SIAM J. Comput.*, 12(4):656–666, 1983.
18. P. Feldman and S. Micali. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM J. Comput.*, 26(4):873–933, 1997.

19. M. Fitzi, N. Gisin, U. M. Maurer, and O. von Rotz. Unconditional byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 482–501. Springer, 2002.
20. M. Fitzi, D. Gottesman, M. Hirt, T. Holenstein, and A. D. Smith. Detectable byzantine agreement secure against faulty majorities. In A. Ricciardi, editor, *Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing, PODC 2002, Monterey, California, USA, July 21-24, 2002*, pages 118–126. ACM, 2002.
21. M. Fitzi and J. B. Nielsen. On the number of synchronous rounds sufficient for authenticated byzantine agreement. In I. Keidar, editor, *Distributed Computing, 23rd International Symposium, DISC 2009, Elche, Spain, September 23-25, 2009. Proceedings*, volume 5805 of *Lecture Notes in Computer Science*, pages 449–463. Springer, 2009.
22. G. Fuchsbauer, J. Katz, and D. Naccache. Efficient rational secret sharing in standard communication networks. In D. Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2010.
23. M. Fujita, T. Koshiba, and K. Yasunaga. Perfectly secure message transmission against rational adversaries. *IEEE J. Sel. Areas Inf. Theory*, 3(2):390–404, 2022.
24. J. A. Garay, J. Katz, C. Koo, and R. Ostrovsky. Round complexity of authenticated broadcast with a dishonest majority. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 658–668. IEEE Computer Society, 2007.
25. J. A. Garay, J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 648–657. IEEE Computer Society, 2013.
26. J. A. Garay, J. Katz, B. Tackmann, and V. Zikas. How fair is your protocol?: A utility-based approach to protocol optimality. In C. Georgiou and P. G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 281–290. ACM, 2015.
27. Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 51–68. ACM, 2017.
28. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In A. V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229. ACM, 1987.
29. S. Goldwasser and Y. Lindell. Secure multi-party computation without agreement. *J. Cryptol.*, 18(3):247–287, 2005.
30. V. Goyal, P. Mohassel, and A. D. Smith. Efficient two party and multi party computation against covert adversaries. In N. P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008.*

Proceedings, volume 4965 of *Lecture Notes in Computer Science*, pages 289–306. Springer, 2008.

31. A. Groce and J. Katz. Fair computation with rational players. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 81–98. Springer, 2012.
32. A. Groce, J. Katz, A. Thiruvengadam, and V. Zikas. Byzantine agreement with a rational adversary. In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 561–572. Springer, 2012.
33. J. Y. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In L. Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 623–632. ACM, 2004.
34. J. Y. Halpern and X. Vilaça. Rational consensus: Extended abstract. In G. Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 137–146. ACM, 2016.
35. R. Hou, H. Yu, and P. Saxena. Using throughput-centric byzantine broadcast to tolerate malicious majority in blockchains. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 1263–1280. IEEE, 2022.
36. J. Katz and C. Koo. On expected constant-round protocols for byzantine agreement. *J. Comput. Syst. Sci.*, 75(2):91–112, 2009.
37. A. Kawachi, Y. Okamoto, K. Tanaka, and K. Yasunaga. General constructions of rational secret sharing with expected constant-round reconstruction. *Comput. J.*, 60(5):711–728, 2017.
38. G. Kol and M. Naor. Games for exchanging information. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 423–432. ACM, 2008.
39. V. Kolesnikov and A. J. Malozemoff. Public verifiability in the covert model (almost) for free. In T. Iwata and J. H. Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 210–235. Springer, 2015.
40. L. Lamport, R. E. Shostak, and M. C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
41. R. Pass and E. Shi. Thunderella: Blockchains with optimistic instant confirmation. In J. B. Nielsen and V. Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 3–33. Springer, 2018.
42. M. C. Pease, R. E. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.
43. T. Rabin. Robust sharing of secrets when the dealer is honest or cheating. *J. ACM*, 41(6):1089–1109, 1994.

44. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In D. S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85. ACM, 1989.
45. P. Scholl, M. Simkin, and L. Siniscalchi. Multiparty computation with covert security and public verifiability. *IACR Cryptol. ePrint Arch.*, 2021:366, 2021.
46. V. Shoup. Practical threshold signatures. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 207–220. Springer, 2000.
47. J. Wan, H. Xiao, S. Devadas, and E. Shi. Round-efficient byzantine broadcast under strongly adaptive and majority corruptions. In R. Pass and K. Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 412–456. Springer, 2020.
48. J. Wan, H. Xiao, E. Shi, and S. Devadas. Expected constant round byzantine broadcast under dishonest majority. In R. Pass and K. Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 381–411. Springer, 2020.
49. A. Yifrach and Y. Mansour. Fair leader election for rational agents in asynchronous rings and networks. In C. Newport and I. Keidar, editors, *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 217–226. ACM, 2018.