Brief Announcement: Single-Round Broadcast: Impossibility, Feasibility, and More

Zhelei Zhou
Zhejiang University, China

Bingsheng Zhang[1]
Zhejiang University, China

Hong-Sheng Zhou
Virginia Commonwealth University, USA

Kui Ren
Zhejiang University, China

Abstract

Broadcast is a fundamental primitive that plays an important role in secure Multi-Party Computation (MPC) area. In this work, we revisit the broadcast with selective abort (hereafter, short for broadcast) proposed by Goldwasser and Lindell (DISC 2002; JoC 2005) and study the *round complexity* of broadcast under different setup assumptions. Our findings are summarized as follows:

- We formally prove that 1-round broadcast is impossible under various widely-used setup assumptions (e.g., plain model, random oracle model, and common reference string model, etc.), even if we consider the static security and the stand-alone framework. More concretely, we formalize a notion called *consistent oracle* to capture these setups, and prove that our impossibility holds under the consistent oracle. Our impossibility holds in both honest majority setting and dishonest majority setting.
- We show that 1-round broadcast protocol is possible in the Universal Composition (UC) framework, by assuming stateful trusted hardwares. Our protocol can be proven secure against all-but-one adaptive and malicious corruptions. We bypass our impossibility result since our stateful trusted hardwares do not satisfy the definition of consistent oracle.
- We provide an application of 1-round broadcast: we construct the first 1-round multiple-verifier zero-knowledge (which is a special case of MPC) protocol, without assuming the broadcast hybrid world.

2012 ACM Subject Classification Security and privacy → Cryptography; Computing methodologies → Distributed algorithms

Keywords and phrases Broadcast; Security with abort; Round optimality

Digital Object Identifier 10.4230/LIPIcs.DISC.2025.62

Category Brief Announcement

Funding *Bingsheng Zhang*: Supported by the National Natural Science Foundation of China (Grant No. 62232002) and Input Output (iohk.io).
Hong-Sheng Zhou: Supported in part by NSF grant CNS-1801470 and a VCU Research Quest grant.

1 Introduction

Broadcast [23] is an important and widely-used cryptographic primitive. Assume there are n parties $P_1, P_2, \ldots, P_n$. Broadcast allows one sending party (say, P_1) to send a message

[1] Bingsheng Zhang and Hong-Sheng Zhou are the corresponding authors.

to the rest receiving parties (say, $P_2, \ldots, P_n$), and the receiving parties should receive the identical messages. Serving as a vital communication channel, broadcast plays an important role in the secure Multi-Party Computation (MPC) area [26, 16], for example, the classic MPC protocol [25] by Rabin and Ben-Or assumes that all parties have access to a broadcast channel. Furthermore, a series of round-optimal MPC protocols [6, 1, 3, 11, 20, 12, 21] are constructed in the broadcast hybrid world.

Round complexity of broadcast. In this work, we focus on the broadcast itself and study how to realize broadcast via only secure point-to-point (P2P) channels. More concretely, we consider *broadcast with selective abort* (hereafter, short for broadcast) proposed by Goldwasser and Lindell [17, 18]: the sending party holds a private input x and sends x to the receiving parties, and each honest receiving party either outputs x or a special symbol $\perp$ indicating abort. We study the *round complexity* of such broadcast protocols under different setup assumptions. Notice that, in this work, we do not consider Byzantine broadcast where the honest receiving parties are guaranteed to receive the same messages.

First of all, we clarify the meaning of "round". In this work, we consider simultaneous communication model, i.e., the parties are allowed to exchange messages in the same round; however, their messages should not depend on each other. For example, in the case of 1-round oblivious transfer protocol [4], the sender and the receiver can send their messages to each other simultaneously and their messages have no dependency; thus, it does not matter which party sends its message first.

Now we introduce the well-known broadcast protocol [17, 18] by Goldwasser and Lindell (hereafter, GL protocol), which is constructed in the plain model and is proven secure against all-but-one malicious corruptions. GL protocol is two-round: in the first round, the sending party sends the message m to the receiving parties; in the second round, the receiving parties send the received message to each other to check if they have received the same message. Since the receiving parties will echo-broadcast the received message from the sending party, GL protocol is also known as the echo-broadcast protocol. GL protocol and its variants are widely used in MPC protocols that are secure with abort.

Intuitively, it seems impossible to construct 1-round broadcast protocol in plain model; to the best of our knowledge, this impossibility result has not yet been formally proven in the literature. In addition to the plain model, there are some commonly-used setup assumptions, e.g., Random Oracle (RO) model, Common Reference String (CRS) model, and ideal cipher model, etc. We wonder if it is possible to construct 1-round broadcast protocol under these setup assumptions. This motivates our main research questions:

Is it possible to construct a 1-round broadcast protocol? If so, under what setup assumptions can such a protocol be constructed?

1.1 Our Results

We investigate the above research questions, and our results are summarized as follows.

Impossibility: 1-round broadcast is impossible under most commonly used setups. We formally prove that 1-round broadcast protocol is impossible under many commonly-used setup assumptions (e.g., RO model, CRS model, ideal cipher model, preprocessing model, etc.). In the main body of this paper, we focus on the Universal Composition (UC) framework [9] and prove the impossibility result under UC framework. Note that, our impossibility result can be extended into the *stand-alone framework* [8, 15].

Feasibility: 1-round broadcast is possible under trusted hardwares. We provide a 1-round broadcast protocol in the *stateful* trusted hardware model, and it is proven to be

UC-secure against an adaptive and rushing adversary who can corrupt up to $t < n$ parties. Furthermore, our broadcast protocol is *non-interactive*: The only communication occurs when the sending party sends the messages to the receiving parties, and the receiving parties do not have to exchange the messages.

Application: 1-round multiple-verifier zero-knowledge. Finally, we present an application of our 1-round broadcast protocol: we construct the first 1-round Multiple-Verifier Zero-Knowledge (MVZK) [7] (a special case of MPC) protocol against a dishonest majority under trusted hardware via only secure P2P channels. Note that, there are three works [24, 2, 27] that claims to achieve 1-round online communication in the preprocessing model; however, they all assume their protocols are in the broadcast hybrid world. Due to the space limit, we leave the details of our MVZK protocol to the full version.

1.2 Related Work

There are limited works that consider broadcast with selective abort. As mentioned in Introduction, Goldwasser and Lindell proposed a 2-round broadcast protocol that is secure with selective abort in the plain model [17, 18] (hereafter, GL protocol). GL protocol is widely-used in MPC with abort (e.g. [13, 5]). In these MPC works, the authors are able to improve the communication complexity of GL protocol by introducing hash function [13] or utilizing the random linear combination technique [5]; however, to the best of knowledge, none of the previous works are able to reduce the round complexity of GL protocol.

2 Preliminaries

2.1 Notation

We denote the security parameter by $\lambda \in \mathbb{N}$. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if for every polynomial $p(\cdot)$ there exists λ_0 such that for all $\lambda > \lambda_0$ we have $\mathsf{negl}(\lambda) < \frac{1}{p(\lambda)}$. We write PPT for a probabilistic polynomial-time algorithm.

2.2 Digital Signature

The digital signature scheme can ensure the integrity of a message. A digital signature scheme (DS) consists of the following algorithms:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{DS.KeyGen}(1^\lambda)$: It takes the security parameter λ as input, and it outputs the public key pk and the signing key sk.
- $\sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}, m)$: It takes the signing key sk and the message m as inputs, and it outputs the signature σ for the message m.
- $\{0,1\} \leftarrow \mathsf{DS.Verify}(\mathsf{pk}, m, \sigma)$: It takes the public key pk, the message m and the signature σ as inputs, and it outputs a bit $b \in \{0,1\}$ indicating the acceptance or rejection.

We require the digital signature scheme to be correct and existentially unforgeable under chosen message attacks (EUF-CMA). Formally, we have the following definitions.

▶ **Definition 1** (Correctness)**.** *We say a digital signature* $\mathsf{DS} = (\mathsf{DS.KeyGen}, \mathsf{DS.Sign}, \mathsf{DS.Verify})$ *is correct if the following equation holds:*

$$\Pr\left[\begin{array}{c}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{DS.KeyGen}(1^\lambda);\\ \sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}, m)\end{array} : \mathsf{DS.Verify}(\mathsf{pk}, m, \sigma) = 1\right] = 1 \ .$$

▶ **Definition 2** (EUF-CMA Security). *We say a digital signature scheme* $\mathsf{DS} = (\mathsf{DS.KeyGen}, \mathsf{DS.Sign}, \mathsf{DS.Verify})$ *is EUF-CMA secure if for any PPT adversary* $\mathcal{A}$*, the following equation holds:*

$$\Pr\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{DS.KeyGen}(1^\lambda); \\ \mathcal{L}_{\mathsf{Sign}} := \emptyset; (m^*,\sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{pk})\end{matrix} : \begin{matrix}\mathsf{DS.Verify}(\mathsf{pk}, m^*, \sigma^*) = 1 \\ \wedge\, m^* \notin \mathcal{L}_{\mathsf{Sign}}\end{matrix}\right] \leq \mathsf{negl}(\lambda) \ ,$$

where $\mathcal{O}_{\mathsf{Sign}}(m)$ *will return* $\sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}, m)$ *and update* $\mathcal{L}_{\mathsf{Sign}} := \mathcal{L}_{\mathsf{Sign}} \cup \{m\}$*.*

2.3 Security Model

In this work, we construct protocols and analyze their security in the Universal Composition (UC) framework by Canetti [9]. We refer readers to see details in [9].

Secure communication model. We consider simultaneous communication [22]: the parties can exchange the messages in the same round, and the parties' messages should have no dependency. We also assume the parties are connected with only secure P2P channels.

Adversarial model. We consider both static and adaptive corruption. We consider a malicious and rushing adversary: the adversary can deviate from the protocol instruction, and it can delay sending messages on behalf of corrupted parties in a given round until the messages sent by all the honest parties in that round have been received. We assume that the adversary is allowed to corrupt the sending party and up to $t < n$ receiving parties, where n is the number of total receiving parties.

2.4 Broadcast Functionality

In this subsection, we provide the ideal functionality for broadcast $\mathcal{F}_{\mathsf{BC}}$ in Figure 1, which is adapted from [18]. The functionality $\mathcal{F}_{\mathsf{BC}}$ interacts with a set of parties $P_1, \ldots, P_n$ and the ideal-world adversary $\mathcal{S}$. Upon receiving the message m from any party, $\mathcal{F}_{\mathsf{BC}}$ will forward the message m to the rest receiving parties; however, $\mathcal{S}$ can cause any honest receiving party to abort. Notice that, in $\mathcal{F}_{\mathsf{BC}}$, $\mathcal{S}$ can make honest receiving parties abort, but $\mathcal{S}$ cannot make honest receiving parties output two distinct messages.

Functionality $\mathcal{F}_{\mathsf{BC}}$

The functionality interacts with a set of parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$. Let $\mathcal{H}$ be the set of honest parties.

Upon receiving $(\textsc{Broadcast}, \mathsf{sid}, m)$ from any party P_i, do:

- Send $(\textsc{Broadcast}, \mathsf{sid}, P_i, m)$ to the adversary $\mathcal{S}$ and the corrupted parties.
- For each honest party $h \in \mathcal{H}$, wait for an input from the adversary $\mathcal{S}$, then do the following:
  - If it is $(\textsc{Continue}, \mathsf{sid}, h)$, send $(\textsc{Broadcast}, \mathsf{sid}, P_i, m)$ to the honest party h.
  - If it is $(\textsc{Abort}, \mathsf{sid}, h)$, send $(\textsc{Abort}, \mathsf{sid})$ to the honest party h.

Figure 1 The Ideal Functionality $\mathcal{F}_{\mathsf{BC}}$

3 Impossibility under Various Setups

In this section, we prove that there exists no 1-round UC-secure broadcast protocol using only *consistent oracles* (to be defined later).

3.1 Consistent Oracles

We introduce the notion of a *consistent oracle* (Definition 3) to model a variety of global-setup behaviours. This notion is adapted from the *monotonically consistent oracle* of [14], which was originally used to establish impossibility results for non-interactive commitment and zero-knowledge protocols in global setups (e.g., the global random oracle) within the generalized UC framework [10].

▶ **Definition 3** (Consistent Oracles). *We say that an "oracle" (or Interactive Turing Machine) is* consistent *if it always returns the same response to the same query made by the same party P in one protocol session.*

▶ Remark 4. The consistent oracle captures a wide range of standard setup assumptions, e.g., the RO model, the CRS model, the PKI model, the preprocessing model, stateless trusted hardware (e.g., signature cards [19]), and combinations thereof.

3.2 Impossibility in the UC Framework

We prove that 1-round one-time broadcast is impossible to achieve using only consistent oracles, even in the presence of a static adversary. We use the method of proof by contradiction to prove the impossibility result. First of all, we assume there exists a 1-round one-time broadcast protocol using only secure P2P channels. Notice that, since we consider the simultaneous communication model, the sending party (say, P_1) is allowed to send messages to the receiving parties (say, $P_2, \ldots, P_n$) via secure P2P channels, and the receiving parties are allowed to communicate with each other, and each parties' messages should not depend on each other. Let us consider the case where the sending party P_1 gets corrupted. Let m, m' be two distinct messages, and let smsg_i (resp. smsg'_i) be the message that an honest sending party should send to the i-th honest receiving party P_i on input m (resp. m'); note that, the message smsg_i (resp. smsg'_i) may contain m (resp. m'). Let $\mathsf{rmsg}_{i,j}$ be the message that an honest receiving party should send to the j-th honest receiving party P_j. Upon receiving smsg_i (resp. smsg'_i) and $\{\mathsf{rmsg}_{i,j}\}_{i\in[2,n]\setminus\{j\}}$, the j-th honest receiving party P_j should output m (resp. m'). With above notations, we can consider the following adversary's strategy: the adversary can instruct the corrupted sending party to query the consistent oracle to prepare $\{\mathsf{smsg}_i\}_{i\in[2,n]}$ and $\{\mathsf{smsg}'_i\}_{i\in[2,n]}$; notice that, the corrupted sending party is capable of doing this by definition of the consistent oracle. Then the adversary can instruct the corrupted sending party to send smsg_2 to the first receiving party P_2, and send $\{\mathsf{smsg}'_j\}_{j\in[3,n]}$ to the rest receiving parties respectively; as a result, P_2 will output m while other receiving parties will output m', which violates the *consensus* property of the one-time broadcast protocol.

Next, we provide the formal theorem statement and the proof is deferred in the full version.

▶ **Theorem 5.** *Let $\mathcal{O}$ denote any PPT consistent oracle as defined in Definition 3. Let n be the number of total parties such that $n \geq 3$. There exists no 1-round one-time broadcast protocol that UC-realizes $\mathcal{F}_{\mathsf{BC}}$ depicted in Figure 1 in the $\mathcal{O}$-hybrid world using only secure P2P channels, in the presence of a static, malicious and rushing adversary who is allowed to corrupt the sending party.*

4 Feasibility under Trusted Hardwares

In this section, we show how to construct 1-round broadcast protocol in the stateful trusted hardware model; notice that, our stateful trusted hardware model does not satisfy the definition of consistent oracle, so we can bypass our impossibility result.

Now we introduce a new trusted hardware model, and we call it *counter-mode* trusted hardware. We also formally define it through an ideal functionality $\mathcal{O}_{\mathsf{HW}}$, which is put in Figure 2. We put our broadcast protocol $\Pi_{\mathsf{BC\text{-}HW}}$ in Figure 3, and state the security of $\Pi_{\mathsf{BC\text{-}HW}}$ through Theorem 6. The proof is deferred in the full version.

Functionality $\mathcal{O}_{\mathsf{HW}}$

It interacts with a set of parties $P_1, \ldots, P_n$ and an adversary $\mathcal{S}$. It is parameterized with $\mathsf{DS} := (\mathsf{DS.KeyGen}, \mathsf{DS.Sign}, \mathsf{DS.Verify})$. It maintains a counter c, which is initialized as 0.

Initialize. Upon receiving $(\textsc{Init}, \mathsf{sid})$ from any party P_i, do:

- Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{DS.KeyGen}(1^\lambda)$ and record $(\mathsf{sid}, \mathsf{pk}, \mathsf{sk})$.
- Ignore any subsequent $\textsc{Init}$ command.

Get public-key. Upon receiving $(\textsc{GetPK}, \mathsf{sid})$ from any party P_i, do:

- If there is a $(\mathsf{sid}, \mathsf{pk}, \mathsf{sk})$ has been recorded, return $(\textsc{GetPK}, \mathsf{sid}, \mathsf{pk})$ to the requester P_i.

Get signature. Upon receiving $(\textsc{Sign}, \mathsf{sid}, m)$ from any party P_i, if there is a $(\mathsf{sid}, \mathsf{pk}, \mathsf{sk})$ has been recorded, generate $\sigma \leftarrow \mathsf{DS.Sign}(\mathsf{sk}, (\mathsf{c}, m))$, return $(\textsc{Sign}, \mathsf{sid}, \sigma)$ to P_i, and increase $\mathsf{c} := \mathsf{c} + 1$.

Figure 2 The Functionality $\mathcal{O}_{\mathsf{HW}}$

Protocol $\Pi_{\mathsf{BC\text{-}HW}}$

Input: P_i privately holds a message x.

Protocol:

- *(Initial phase)*: The parties send $(\textsc{Init}, \mathsf{sid})$ to $\mathcal{O}_{\mathsf{HW}}$. Each party P_j initializes a counter $\mathsf{c}_j := 0$ for $j \in [n]$.
- *(P_i broadcasts the message)*: P_i sends $(\textsc{Sign}, \mathsf{sid}, m)$ to $\mathcal{O}_{\mathsf{HW}}$, which returns $(\textsc{Sign}, \mathsf{sid}, \sigma)$. Then P_i increases its counter $\mathsf{c}_i := \mathsf{c}_i + 1$ and sends (m, σ) to the rest parties via secure P2P channels.
- *(P_j locally checks the validity of the received message)*: For each $j \in [n] \setminus \{i\}$, P_j sends $(\textsc{GetPK}, \mathsf{sid})$ to $\mathcal{O}_{\mathsf{HW}}$, which returns $(\textsc{GetPK}, \mathsf{sid}, \mathsf{pk})$. Then P_j checks if $\mathsf{DS.Verify}(\mathsf{pk}, (\mathsf{c}_j, m), \sigma) = 1$ holds. If so, P_j accepts m as the received broadcast message and increases its counter $\mathsf{c}_j := \mathsf{c}_j + 1$; otherwise, P_j aborts.

Figure 3 1-round broadcast protocol $\Pi_{\mathsf{BC\text{-}HW}}$ in the $\mathcal{O}_{\mathsf{HW}}$-hybrid world

▶ **Theorem 6.** *Assume* $\mathsf{DS} = (\mathsf{DS.KeyGen}, \mathsf{DS.Sign}, \mathsf{DS.Verify})$ *is a digital signature scheme that is correct and EUF-CMA secure. The protocol* $\Pi_{\mathsf{BC\text{-}HW}}$ *depicted in Figure 3 UC-realizes the functionality* $\mathcal{F}_{\mathsf{BC}}$ *depicted in Figure 1 in the* $\mathcal{O}_{\mathsf{HW}}$*-hybrid world, in the presence of an adaptive and malicious adversary who can corrupt up to* $t < n$ *parties.*

References

1 Prabhanjan Ananth, Arka Rai Choudhuri, and Abhishek Jain. A new approach to round-optimal secure multiparty computation. In *CRYPTO 2017*.

2 Benny Applebaum, Eliran Kachlon, and Arpita Patra. Verifiable relation sharing and multi-verifier zero-knowledge in two rounds: Trading NIZKs with honest majority - (extended abstract). In *CRYPTO 2022*.
3 Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal MPC. In *CRYPTO 2018*.
4 Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In *CRYPTO 1989*.
5 Elette Boyle, Niv Gilboa, Yuval Ishai, and Ariel Nof. Sublinear GMW-style compiler for MPC with preprocessing. In *CRYPTO 2021*.
6 Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. In *TCC 2017*.
7 Mike Burmester and Yvo Desmedt. Broadcast interactive proofs (extended abstract). In *EUROCRYPT 1991*.
8 Ran Canetti. Security and composition of multiparty cryptographic protocols. In *Journal of Cryptology 2000*.
9 Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS 2001*.
10 Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In *TCC 2007*.
11 Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Round optimal secure multiparty computation from minimal assumptions. In *TCC 2020*.
12 Michele Ciampi, Divya Ravi, Luisa Siniscalchi, and Hendrik Waldner. Round-optimal multi-party computation with identifiable abort. In *EUROCRYPT 2022*.
13 Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO 2012*.
14 Yevgeniy Dodis, Victor Shoup, and Shabsi Walfish. Efficient constructions of composable commitments and zero-knowledge proofs. In *CRYPTO 2008*.
15 Oded Goldreich. Foundations of cryptography, volume 2.
16 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC 1987*.
17 Shafi Goldwasser and Yehuda Lindell. Secure computation without agreement. In *DISC 2002*.
18 Shafi Goldwasser and Yehuda Lindell. Secure multi-party computation without agreement. In *Journal of Cryptology 2005*.
19 Dennis Hofheinz, Jörn Müller-Quade, and Dominique Unruh. Universally composable zero-knowledge arguments and commitments from signature cards. In *Central European Conference on Cryptology 2005*.
20 Yuval Ishai, Dakshita Khurana, Amit Sahai, and Akshayaram Srinivasan. On the round complexity of black-box secure MPC. In *CRYPTO 2021*.
21 Yuval Ishai, Dakshita Khurana, Amit Sahai, and Akshayaram Srinivasan. Round-optimal black-box MPC in the plain model. In *CRYPTO 2023*.
22 Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Universally composable synchronous computation. In *TCC 2013*.
23 Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. In ACM Trans. Program. Lang. Syst. 1982.
24 Matt Lepinski, Silvio Micali, and Abhi Shelat. Fair-zero knowledge. In *TCC 2005*.
25 Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *STOC 1989*.
26 Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS 1982*.
27 Zhelei Zhou, Bingsheng Zhang, Hong-Sheng Zhou, and Kui Ren. Single-input functionality against a dishonest majority: Practical and round-optimal. In *PKC 2025*.