

Broadcast from Minicast Secure Against General Adversaries^{*}

Pavel Raykov^{1,2,**}

¹ School of Electrical Engineering, Tel-Aviv University, Israel

² ITMO University, 49 Kronverkskiy av., Saint-Petersburg, Russia, 197101

pavelraykov@post.tau.ac.il

Abstract. Byzantine broadcast is a distributed primitive that allows a specific party to consistently distribute a message among n parties in the presence of potential misbehavior of up to t of the parties. The celebrated result of [PSL80] shows that broadcast is achievable from point-to-point channels if and only if $t < n/3$.

The following two generalizations have been proposed to the original broadcast problem. In [FM98] the authors considered a *general adversary* characterized by the sets of parties that can be corrupted. It was shown that broadcast is achievable from point-to-point channels if and only if no three possible corrupted sets can cover the whole party set. In [CFF⁺05] the notion of point-to-point channels has been extended to the b -minicast channels allowing to locally broadcast among any subset of b parties. It has been shown that broadcast secure against adversaries corrupting up to t parties is achievable from b -minicast if and only if $t < \frac{b-1}{b+1}n$.

In this paper we combine both generalizations by considering the problem of achieving broadcast from b -minicast channels secure against general adversaries. Our main result is a condition on the possible corrupted sets such that broadcast is achievable from b -minicast if and only if this condition holds.

1 Introduction

1.1 Byzantine Broadcast

The Byzantine broadcast problem (aka Byzantine generals) is formulated as follows [PSL80]: A specific party (the sender) wants to distribute a message among n parties in such a way that all correct parties obtain the same message, even when some of the parties are malicious. The malicious misbehavior is modeled by a central adversary corrupting up to t parties and taking full control of their actions. Corrupted parties are called *Byzantine* and the remaining parties are called *correct*. Broadcast requires that all correct parties agree on the same value v , and if the sender is correct, then v is the value proposed by the sender. Broadcast is one of the most fundamental primitives in distributed computing. It is used to implement various protocols like voting, bidding, collective contract signing, etc. Basically, this list can be continued with all protocols for secure multi-party computation (MPC) as defined in [Yao82, GMW87].

There exist various implementations of Byzantine broadcast from synchronous point-to-point communication channels with different security guarantees. In the model without trusted setup, perfectly-secure Byzantine broadcast is achievable if and only if $t < n/3$ [PSL80, BGP92, CW92]. In the model with trusted setup, information-theoretically or cryptographically secure Byzantine broadcast is achievable for any $t < n$ [DS83, PW96].

^{*} This is the full version of the paper appearing at ICALP 2015.

^{**} Supported by ISF grant 1155/11, Israel Ministry of Science and Technology (grant 3-9094), GIF grant 1152/2011, and the Check Point Institute for Information Security.

1.2 Extending the Broadcast Problem

We consider the following two extensions of the broadcast problem.

b -Minicast Communication Model. The classical Byzantine broadcast problem [PSL80] assumes that the parties communicate with point-to-point channels only. Fitzi and Maurer [FM00] considered a model where the parties can additionally access partial broadcast channels among any set of b parties. We call such a partial broadcast channel b -minicast. Then, one can interpret the point-to-point communication model as the 2-minicast model. Hence, the result of [PSL80] states that broadcast is achievable from 2-minicast if and only if the number of corrupted parties $t < n/3$. In [FM00] it is shown that broadcast is achievable from 3-minicast if and only if $t < n/2$. Later this was generalized for the arbitrary b -minicast model—in [CFF⁺05] it is proved that broadcast is achievable from b -minicast if and only if $t < \frac{b-1}{b+1}n$.

It has also been studied how many 3-minicast channels need to be available in order to achieve broadcast [JMS12]. A general MPC protocol in the asynchronous model using 3-minicasts has been considered in [BBCK14].

General Adversaries. Originally, the broadcast problem has been stated for a *threshold* adversary that can corrupt any set of parties A such that $|A| \leq t$ for some threshold t . Fitzi and Maurer [FM98] considered a model with a *general* adversary that can corrupt a set of parties A such that $A \in \mathcal{A}$ for some family of possible corrupted sets \mathcal{A} . In [FM98] it has been shown that broadcast is achievable from point-to-point channels if and only if the adversary cannot corrupt three sets of parties that cover the whole party set.

A MPC protocol secure against general adversaries is given in [HM97]. More efficient MPC protocols secure against general adversaries are studied in [Mau02, HMZ08, HT13, LO14].

One of the most prominent approaches to construction of protocols secure against general adversaries is the “player emulation” technique of [HM97]. Its main idea is a generation of a new set of “virtual parties” \mathcal{V} that are emulated by the original parties \mathcal{P} . Then the problem of constructing a protocol among \mathcal{P} is reduced to a protocol construction among \mathcal{V} . As an example application of the player emulation technique, consider the protocols by [RVS⁺04, CDI⁺13]. These protocols construct broadcast from 3-minicast and tolerate any general adversary who cannot corrupt two sets of parties that cover the whole party set. In [RVS⁺04, CDI⁺13], triples of actual parties from \mathcal{P} are used to emulate virtual parties, where the emulation protocol is implemented with the help of 3-minicast.

1.3 Contributions

We consider the combination of the two described extensions for the broadcast problem. That is, we study which general adversaries can be tolerated while constructing broadcast from b -minicast channels. We completely resolve this question by **(1)** giving a condition on a general adversary that can be tolerated while implementing broadcast from b -minicast and **(2)** showing that this condition is tight. Our results improve the previous work on generalized adversaries in the minicast communication model [FM00, RVS⁺04, CDI⁺13]. For example, consider a setting with 4 parties P_1, P_2, P_3, P_4 in the 3-minicast model. In Table 1 we illustrate for which general adversaries our results are new.

To show **(1)** we construct a protocol that realizes broadcast from minicast and is secure against general adversaries. The protocol we give does not employ the player emulation technique and is inspired by the original protocol of [CFF⁺05] that is secure against threshold adversaries. To show **(2)** we reduce a protocol secure against a general adversary to a protocol that is secure against a threshold adversary such that the former is impossible according to [CFF⁺05].

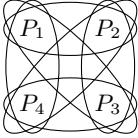
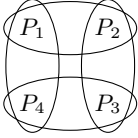
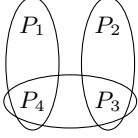
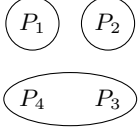
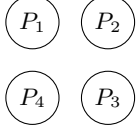
Possible corrupted sets	Broadcast possible?	Literature
	No	[FM00]
	No	This work
	Yes	This work
	Yes	[RVS ⁺ 04, CDI ⁺ 13]
	Yes	[FM00]

Table 1. The overview of tolerable general adversaries in the 3-minicast model. In the first column each maximal corrupted set is shown with an oval. In the third column we give a reference to the first work to show whether one can construct broadcast from 3-minicasts in the setting where any of the possible corrupted sets can be controlled by the adversary.

2 Model and Definitions

Parties. We consider a setting with n parties (also called players) $\mathcal{P} = \{P_1, \dots, P_n\}$. We employ the following notion of party set partition:

Definition 1. A list $\mathcal{S} = (S_0, \dots, S_{k-1})$ is a k -partition of \mathcal{P} if $\bigcup_{i=0}^{k-1} S_i = \mathcal{P}$ and all S_i, S_j are pair-wise disjoint. Furthermore, if all S_i are non-empty we call such a partition proper.

We introduce additional notation to denote the set of parties from \mathcal{P} without two sets S_i and S_j from the partition \mathcal{S} : Let $\mathcal{S}_{\downarrow i,j} := \mathcal{P} \setminus (S_{i \bmod k} \cup S_{j \bmod k})$.

Adversary. We assume that some of the parties can be corrupted by a central adversary making them deviate from the prescribed protocol in any desired manner. Before the execution of the protocol the adversary must specify the set of parties $A \subseteq \mathcal{P}$ to corrupt. The choice of the adversary is limited by means of a family of possible corrupted sets $\mathcal{A} \subseteq 2^{\mathcal{P}}$, i.e., the corrupted set A can be chosen only from \mathcal{A} . We assume that \mathcal{A} is monotone, i.e., for $\forall S, S' (S \in \mathcal{A}) \wedge (S' \subseteq S) \Rightarrow S' \in \mathcal{A}$. The set \mathcal{A} is called an *adversarial structure*. We consider *perfect security* which captures the fact that the protocol never fails even if the adversary has unbounded computing power.

The parties that are corrupted are also called *Byzantine* or *malicious*, while the remaining uncorrupted parties are called *correct* or *honest*.

Communication. In the classical setting [PSL80], it is assumed that the parties are connected with a synchronous authenticated point-to-point network. Synchronous means that all parties share a common clock and that the message delay in the network is bounded by a constant. In this paper we consider an extended model where messages can be consistently delivered to more than one recipient via a b -minicast channel.

Definition 2 (b -minicast model). *In the b -minicast model, in every subset $Q \subseteq \mathcal{P}$ of at most b parties each party $P_i \in Q$ has access to a b -minicast channel that takes an input v from some domain \mathcal{D} from P_i and outputs v to all parties in Q . Each of the b -minicast channels is also synchronous, i.e., the message delivery delay to all recipients is bounded by a constant.*

Note that the classical setting with point-to-point channels can be seen as an instantiation of the b -minicast model for $b = 2$.

Broadcast protocols. A broadcast protocol allows parties to simulate a global broadcast channel with the help of the communication means available in the presence of an adversary. Formally, this is defined as follows.

Definition 3 (Broadcast). *A protocol among the parties \mathcal{P} where some party $P_s \in \mathcal{P}$ (called the sender) holds an input $v \in \mathcal{D}$ and every party $P_i \in \mathcal{P}$ outputs a value $y_i \in \mathcal{D}$ achieves broadcast if the following holds:*

VALIDITY: *If the sender P_s is correct, then every correct party $P_i \in \mathcal{P}$ outputs the sender's value $y_i = v$.*

CONSISTENCY: *All correct parties in \mathcal{P} output the same value.*

3 The Main Result

We now characterize which adversary structures \mathcal{A} can be tolerated while implementing broadcast from b -minicast channels. Our condition is inspired by the impossibility proof of [CFF⁺05]. There it is shown that no protocol can realize broadcast among $b + 1$ parties from b -minicast where the adversary can corrupt any number of parties. The impossibility proof of [CFF⁺05] is built by considering a chain of parties P_1, \dots, P_{b+1} where any pair of parties $P_i, P_{i+1 \bmod b+1}$ ($i = 1, \dots, b+1$) can be honest, while the remaining parties are corrupted. We generalize this to a chain of party sets:

Definition 4. *A structure \mathcal{A} is said to contain a k -chain if there exists a proper k -partition $\mathcal{S} = (S_0, \dots, S_{k-1})$ of \mathcal{P} such that $\forall i \in [0, k-1] \mathcal{S}_{\downarrow i, i+1} \in \mathcal{A}$. A structure is called k -chain-free if it does not have a k -chain.*

Our main result can be formulated as follows.

Theorem 1. *In the b -minicast communication model, broadcast tolerating adversary structure \mathcal{A} is achievable if and only if \mathcal{A} is $(b + 1)$ -chain-free.*

The proof of the theorem is split into two parts. In Section 4 we give a protocol that realizes broadcast from b -minicast channels and tolerates any $(b + 1)$ -chain-free \mathcal{A} . In Section 5 we show that no protocol can implement broadcast in this model while tolerating some \mathcal{A} that has a $(b + 1)$ -chain.

4 The Feasibility Proof

In this section we construct a broadcast protocol that uses b -minicast channels and tolerates any adversarial structure \mathcal{A} which is $(b + 1)$ -chain-free. Our construction consists of three steps and is based on [CFF⁺05]. First, we introduce a new distributed primitive called proxcast and show how to realize proxcast from b -minicast while tolerating arbitrary corruptions. Second, we consider a broadcast primitive with hybrid security, i.e., depending on which set of parties the adversary corrupts, the primitive satisfies only one of the broadcast security guarantees. We then show how to implement such a hybrid broadcast primitive from proxcast. Finally, given broadcast with hybrid security, we implement broadcast secure against any \mathcal{A} which is $(b + 1)$ -chain-free.

4.1 Proxcast

Proxcast is a relaxed version of the binary broadcast primitive in that it has a weakened consistency property. As a result of a proxcast invocation, each of the parties P_i outputs a level ℓ_i from some range $[0, \ell - 1]$ indicating whether 0 was likely to be proxcast (lower levels) or 1 (higher levels). It is guaranteed that if the sender is correct, then any correct P_i outputs $\ell_i = 0$ if 0 is proxcast, and $\ell_i = \ell - 1$ if 1 is proxcast. The consistency property of proxcast guarantees that levels of correct parties are close, however, they may be different. Formally, we define proxcast as follows.

Definition 5 (ℓ -proxcast). *Let $\ell \in \mathbb{N}$. A protocol among \mathcal{P} where the sender $P_s \in \mathcal{P}$ holds an input $v \in \{0, 1\}$ and every party $P_i \in \mathcal{P}$ finally outputs a level $\ell_i \in [0, \ell - 1]$, achieves ℓ -proxcast if the following holds:*

VALIDITY: *If P_s is correct then all correct P_i output $\ell_i = (\ell - 1)v$.*

CONSISTENCY: *There exists $k \in [0, \ell - 2]$ such that each correct P_i outputs level $\ell_i \in \{k, k + 1\}$.*

Our construction of b -proxcast from b -minicasts is a simplification of the proxcast protocol of [CFF⁺05]. We let the sender P_s b -minicast the value he holds among all subsets of b parties. Then each P_i computes the level ℓ_i to be the minimum number of parties with whom P_i sees only zeros.

Protocol $\text{Proxcast}_b(\mathcal{P}, P_s, v)$

1. If $|\mathcal{P}| \leq b$ then broadcast v using b -minicast. Let y_i denote the output of P_i . Each P_i decides on $\ell_i := (b - 1)y_i$.
2. Otherwise:
 - 2.1 Let $\mathcal{B} = \{S \subseteq \mathcal{P} \mid (P_s \in S) \wedge (|S| = b)\}$.
 $\forall S \in \mathcal{B}$: P_s b -minicasts v among S . Let y_S be the output of each $P_i \in S$.
 - 2.2 $\forall P_i \in \mathcal{P} \setminus \{P_s\}$: For each $T \subseteq \mathcal{P} \setminus \{P_s, P_i\}$ with $|T| \leq b - 2$, let $V_i^T := \{y_S \mid (S \in \mathcal{B}) \wedge (P_i \in S) \wedge (T \subseteq S)\}$. Output ℓ_i to be the minimum $|T|$ such that $V_i^T = \{0\}$ (if no such T exists output $\ell_i := b - 1$).
The sender P_s : Output $\ell_s := v(b - 1)$.

Lemma 1. *In the b -minicast model, the protocol Proxcast_b perfectly securely achieves b -proxcast in the presence of any adversary.*

Proof. If $|\mathcal{P}| \leq b$ then the protocol trivially achieves Proxcast . Assume now that $|\mathcal{P}| > b$. We show that each of the proxcast properties is satisfied:

VALIDITY: If the sender P_s is honest then P_s b -minicasts only v . Hence any correct P_i computes all V_i^T to be $\{v\}$ at Step 2.2. If $v = 0$, then V_i^\emptyset is $\{0\}$ and each correct P_i outputs $\ell_i = |\emptyset| = 0$. If $v = 1$, then all V_i^T are $\{1\}$ and hence no T with $V_i^T = \{0\}$ exists. Consequently, each correct party P_i outputs $\ell_i = b - 1$.

CONSISTENCY: First, if the sender is correct then the consistency property trivially holds. Assume now that the sender is corrupted. Let $P_i \in \mathcal{P} \setminus \{P_s\}$ denote the party outputting the smallest ℓ_i among all honest receivers. Let T_i be the set with $|T_i| = \ell_i$ and $V_i^{T_i} = \{0\}$. Take any other correct $P_j \in \mathcal{P} \setminus \{P_s\}$. We show that $\ell_j \leq \ell_i + 1$. If $\ell_i \in \{b - 2, b - 1\}$ then this trivially holds. Assume now that $\ell_i \leq b - 3$. Consider two cases:

($P_j \notin T_i$) Let $T' := T_i \cup \{P_i\}$. We have that any $y_S = 0$ if $T' \subseteq S$ and $P_j \in S$. Hence, $V_j^{T'} = \{0\}$.

Consequently, $\ell_j \leq |T'| = \ell_i + 1$.

($P_j \in T_i$) Let $T' := T_i \cup \{P_i\} \setminus \{P_j\}$. We have that any $y_S = 0$ if $T' \subseteq S$ and $P_j \in S$. Hence, $V_j^{T'} = \{0\}$. Consequently, $\ell_j \leq |T'| = \ell_i$. \square

4.2 Broadcast with Hybrid Security

In [FHHW03] the authors proposed a more fine-grained security definition of broadcast in the setting with a threshold adversary. The new primitive is called a two-threshold broadcast. In the two-threshold version of broadcast, the consistency and validity properties of broadcast are guaranteed to be achieved in the presence of up to t_c and t_v malicious parties, respectively. It is shown in [FHHW03] that one can implement two-threshold broadcast from point-to-point channels if and only if $t = 0$ or $t + 2T < n$ (where $t = \min(t_c, t_v)$ and $T = \max(t_c, t_v)$).

We consider an extended notion of the two-threshold broadcast with respect to general adversaries. We call the resulting primitive hybrid broadcast.³

Definition 6 (Hybrid broadcast). *A protocol among \mathcal{P} where the sender $P_s \in \mathcal{P}$ holds an input $v \in \mathcal{D}$ and every party $P_i \in \mathcal{P}$ finally outputs a value $y_i \in \mathcal{D}$, achieves hybrid broadcast with respect to a pair of adversarial structures $(\mathcal{A}_c, \mathcal{A}_v)$ if the following holds:*

VALIDITY: *If the sender P_s is honest and a set $A \in \mathcal{A}_v$ of parties is corrupted then all honest $P_i \in \mathcal{P} \setminus A$ output sender's value $y_i = v$.*

CONSISTENCY: *If a set $A \in \mathcal{A}_c$ of parties is corrupted then all honest $P_i \in \mathcal{P} \setminus A$ output the same value.*

We construct a hybrid broadcast protocol in two steps. First, we introduce additional operations on adversary structures and prove properties of them. Then, we present the protocol.

Additional tools. We start by defining two operators **del** and **proj** that transform adversarial structures. The first operator **del** chooses only the sets $A \in \mathcal{A}$ that do not contain a specific party P_i . The second operator **proj** selects only the sets $A \in \mathcal{A}$ that can be corrupted together with a specific party P_i . Formally, $\text{del}(\mathcal{A}, P_i) := \{A \in \mathcal{A} \mid P_i \notin A\}$ and $\text{proj}(\mathcal{A}, P_i) := \{A \in \text{del}(\mathcal{A}, P_i) \mid A \cup \{P_i\} \in \mathcal{A}\}$. We prove that $\text{del}(\mathcal{A}, P_i)$ and $\text{proj}(\mathcal{A}, P_i)$ are adversarial structures over $\mathcal{P} \setminus \{P_i\}$.

³ Note that our extension for non-threshold adversaries is more general than the one given in [FHHW03], because we allow any \mathcal{A}_c and \mathcal{A}_v , and not only $\mathcal{A}_c \subseteq \mathcal{A}_v$ or $\mathcal{A}_v \subseteq \mathcal{A}_c$ as [FHHW03]. This difference stems from the fact that we treat validity and consistency equally, while [FHHW03] considers the case where the broadcast security properties can be “degraded” if the adversarial power grows. That is, if the adversary corrupts A from some \mathcal{A}_{small} then the broadcast protocol must satisfy both validity and consistency, while if the adversary corrupts A from some $\mathcal{A}_{big} \supseteq \mathcal{A}_{small}$, then only validity (or consistency) is required to be satisfied.

Lemma 2. *If \mathcal{A} is an adversarial structure over \mathcal{P} , then $\text{del}(\mathcal{A}, P_i)$ and $\text{proj}(\mathcal{A}, P_i)$ are adversarial structures over $\mathcal{P} \setminus \{P_i\}$.*

Proof. By definition we have that $\text{del}(\mathcal{A}, P_i)$ and $\text{proj}(\mathcal{A}, P_i)$ are subsets of $2^{\mathcal{P} \setminus \{P_i\}}$. We now prove that they are monotone.

Take any $A \in \text{del}(\mathcal{A}, P_i)$ and $A' \subseteq A$. Because $P_i \notin A$ we have that $P_i \notin A'$, hence $A' \in \text{del}(\mathcal{A}, P_i)$.

Take any $B \in \text{proj}(\mathcal{A}, P_i)$ and $B' \subseteq B$. Because $B \in \text{del}(\mathcal{A}, P_i)$ we have that $B' \in \text{del}(\mathcal{A}, P_i)$. Consider $B' \cup \{P_i\}$. Because $B' \cup \{P_i\} \subseteq B \cup \{P_i\}$, $B \cup \{P_i\} \in \mathcal{A}$ and \mathcal{A} is monotone, we have that $B' \cup \{P_i\} \in \mathcal{A}$. Hence, $B' \in \text{proj}(\mathcal{A}, P_i)$. \square

We now give a condition on adversarial structures $(\mathcal{A}_c, \mathcal{A}_v)$, such that if this condition holds then there exists a protocol that constructs hybrid broadcast from b -minicasts and tolerates $(\mathcal{A}_c, \mathcal{A}_v)$. Similarly to the k -chain-free condition, it is inspired by the chain of parties considered in the impossibility proof of [CFF⁺05].

Definition 7. *A pair of structures $(\mathcal{A}_c, \mathcal{A}_v)$ is said to be k -chain-free-compatible if for any proper k -partition \mathcal{S} of \mathcal{P} , there exists $i \in [0, k-3]$ such that $\mathcal{S}_{\downarrow i, i+1} \notin \mathcal{A}_c$ or $\mathcal{S}_{\downarrow k-2, k-1} \notin \mathcal{A}_v$ or $\mathcal{S}_{\downarrow k-1, 0} \notin \mathcal{A}_v$.*

We show that k -chain-free-compatible structures satisfy the following useful properties:

Lemma 3. *Let $(\mathcal{A}_c, \mathcal{A}_v)$ be k -chain-free-compatible and $P_i \in \mathcal{P}$. Then the pair $(\text{proj}(\mathcal{A}_c, P_i), \text{del}(\mathcal{A}_v, P_i))$ is also k -chain-free-compatible.*

Proof. Let $\mathcal{A}'_c := \text{proj}(\mathcal{A}_c, P_i)$ and $\mathcal{A}'_v := \text{del}(\mathcal{A}_v, P_i)$. Assume for contradiction that $(\mathcal{A}'_c, \mathcal{A}'_v)$ is not k -chain-free-compatible. Then, there exists a proper k -partition $\mathcal{S} = (S_0, \dots, S_{k-1})$ of $\mathcal{P} \setminus \{P_i\}$ such that the formula

$$F = \left(\bigwedge_{j=0}^{k-3} \mathcal{S}_{\downarrow j, j+1} \in \mathcal{A}'_c \right) \wedge (\mathcal{S}_{\downarrow k-2, k-1} \in \mathcal{A}'_v) \wedge (\mathcal{S}_{\downarrow k-1, 0} \in \mathcal{A}'_v)$$

is true. Define a new system of sets R_0, \dots, R_{k-1} as following: for all $j \in [0, k-2]$ let $R_j := S_j$ and let $R_{k-1} := S_{k-1} \cup \{P_i\}$. By construction we have that $\mathcal{R} = (R_0, \dots, R_{k-1})$ is a proper k -partition of \mathcal{P} . We now show that

$$G = \left(\bigwedge_{j=0}^{k-3} \mathcal{R}_{\downarrow j, j+1} \in \mathcal{A}_c \right) \wedge (\mathcal{R}_{\downarrow k-2, k-1} \in \mathcal{A}_v) \wedge (\mathcal{R}_{\downarrow k-1, 0} \in \mathcal{A}_v)$$

is true, meaning that $(\mathcal{A}_c, \mathcal{A}_v)$ is not k -chain-free-compatible. This will be the contradiction.

First, given that F is true, we have that $\mathcal{S}_{\downarrow k-2, k-1}, \mathcal{S}_{\downarrow k-1, 0} \in \mathcal{A}'_v$. By the construction of \mathcal{R} , we have that $\mathcal{R}_{\downarrow k-2, k-1} = \mathcal{S}_{\downarrow k-2, k-1}$ and $\mathcal{R}_{\downarrow k-1, 0} = \mathcal{S}_{\downarrow k-1, 0}$. Because $\mathcal{S}_{\downarrow k-2, k-1}, \mathcal{S}_{\downarrow k-1, 0} \in \mathcal{A}'_v$, we get that $\mathcal{R}_{\downarrow k-2, k-1}$ and $\mathcal{R}_{\downarrow k-1, 0}$ are from \mathcal{A}'_v . Since $\mathcal{A}'_v = \text{del}(\mathcal{A}_v, P_i)$ it holds that $\mathcal{A}'_v \subseteq \mathcal{A}_v$. Consequently, $\mathcal{R}_{\downarrow k-2, k-1}, \mathcal{R}_{\downarrow k-1, 0} \in \mathcal{A}_v$.

Second, since F is true, we have that for all $j \in [0, k-3]$ it holds that $\mathcal{S}_{\downarrow j, j+1} \in \mathcal{A}'_c$. Because $\mathcal{A}'_c = \text{proj}(\mathcal{A}_c, P_i)$ it must hold that $\mathcal{R}_{\downarrow j, j+1} = \mathcal{S}_{\downarrow j, j+1} \cup \{P_i\} \in \mathcal{A}_c$. This implies that for all $j \in [0, k-3]$ we have that $\mathcal{R}_{\downarrow j, j+1} \in \mathcal{A}_c$.

Finally, we conclude that G is true. \square

Lemma 4. *If \mathcal{A} is k -chain-free, then $(\mathcal{A}, \mathcal{A})$ is k -chain-free-compatible.*

Proof. Consider any proper k -partition \mathcal{S} of \mathcal{P} . We have that

$$\left(\bigwedge_{i=0}^{k-3} \mathcal{S}_{\downarrow i, i+1} \in \mathcal{A} \right) \wedge (\mathcal{S}_{\downarrow k-2, k-1} \in \mathcal{A}) \wedge (\mathcal{S}_{\downarrow k-1, 0} \in \mathcal{A}) \equiv \bigwedge_{i=0}^{k-1} \mathcal{S}_{\downarrow i, i+1} \in \mathcal{A}.$$

This means that to test whether \mathcal{A} is k -chain-free or $(\mathcal{A}, \mathcal{A})$ is k -chain-free-compatible we check the same condition. Hence, if \mathcal{A} is k -chain-free, then $(\mathcal{A}, \mathcal{A})$ is k -chain-free-compatible. \square

The protocol. The protocol `HybridBC` works recursively as follows. First, the sender P_s proxcasts the value v he holds to everyone in \mathcal{P} . Then, each of the receivers in $\mathcal{P} \setminus \{P_s\}$ invokes `HybridBC` again to hybridly broadcast his level among the receivers. Now the view of every receiver P_i consists of $n - 1$ levels of the others. Then, P_i partitions the receiver set $\mathcal{P} \setminus \{P_s\}$ into b subsets according to the level that is hybridly broadcast by the parties in the set. By analyzing the properties of this partition, each P_i takes his final decision.

Protocol HybridBC($\mathcal{P}, P_s, v, \mathcal{A}_c, \mathcal{A}_v$)

1. If $|\mathcal{P}| \leq b$ then broadcast v using b -minicast.
2. Otherwise:

2.1 Parties in \mathcal{P} invoke `Proxcastb`(\mathcal{P}, P_s, v). Let ℓ_i denote the output of P_i .

2.2 Let $\mathcal{A}'_c := \text{proj}(\mathcal{A}_c, P_s)$, $\mathcal{A}'_v := \text{del}(\mathcal{A}_v, P_s)$ and $\mathcal{P}' := \mathcal{P} \setminus \{P_s\}$.

2.3 $\forall P_j \in \mathcal{P}'$: Parties in \mathcal{P}' invoke `HybridBC`($\mathcal{P}', P_j, \ell_j, \mathcal{A}'_c, \mathcal{A}'_v$).⁴

Let ℓ'_j denote the output of P_j .

2.4 $\forall P_i \in \mathcal{P}'$: For all $\ell \in [0, b - 1]$ let $L_\ell^i := \{P_j \in \mathcal{P}' \mid \ell'_j = \ell\}$. Let $L_b^i = \{P_s\}$. Let $\mathcal{L}^i = (L_0^i, \dots, L_b^i)$ be a $(b + 1)$ -partition of \mathcal{P} .

2.5 $\forall P_i \in \mathcal{P}'$: Output 0 if

$$(\mathcal{L}_{\downarrow b, 0}^i \in \mathcal{A}_v) \wedge \left(\bigwedge_{k=0}^{\ell_i} L_k^i \neq \emptyset \right) \wedge \left(\bigwedge_{k=0}^{\ell_i-1} \mathcal{L}_{\downarrow k, k+1}^i \in \mathcal{A}_c \right).⁵$$

Otherwise, output 1.

2.6 The sender P_s outputs v .

Lemma 5. *In the b -minicast model, the protocol `HybridBC` perfectly securely achieves binary hybrid broadcast if the pair $(\mathcal{A}_c, \mathcal{A}_v)$ is $(b + 1)$ -chain-free-compatible.*

Proof. The proof proceeds by induction over the size of the party set $|\mathcal{P}| = n$. If $|\mathcal{P}| \leq b$ then broadcast is directly achieved with the help of b -minicast. Now we prove the induction step. We assume that the protocol `HybridBC` achieves hybrid broadcast for any set of $n - 1$ parties.

Consider now any $(b + 1)$ -chain-free-compatible $(\mathcal{A}_c, \mathcal{A}_v)$ over the set of n parties \mathcal{P} . Due to Lemma 3, the pair $(\mathcal{A}'_c, \mathcal{A}'_v)$ computed at Step 2.2 is $(b + 1)$ -chain-free-compatible. Hence, we can assume that each recursive invocation of the protocol `HybridBC` at Step 2.3 achieves hybrid broadcast.

Now we prove each of the hybrid broadcast security properties:

⁴ The protocol `HybridBC` works for binary values only. This invocation is translated into $\log b$ parallel invocations of `HybridBC` to broadcast ℓ_j bit by bit.

⁵ We assume that $\bigwedge_{k=0}^{-1}(\dots)$ is always true.

VALIDITY: We assume that the sender P_s is correct, and we show that all correct parties in \mathcal{P} output v . Because P_s always outputs v , we are left to show only that all correct receivers in $\mathcal{P}' = \mathcal{P} \setminus \{P_s\}$ output v . Assume that the adversary corrupts some $A \in \mathcal{A}_v$. Let $H = \mathcal{P}' \setminus A$ denote the set of the remaining honest receivers. We consider two cases:

- ($v = 0$) Because P_s is correct, Proxcast_b guarantees that every $P_i \in H$ has $\ell_i = 0$. Now consider any correct P_x . We have that $H \subseteq L_0^x$. Because \mathcal{A}_v is monotone and $\mathcal{P}' \setminus H \in \mathcal{A}_v$, we get that $\mathcal{P}' \setminus L_0^x \in \mathcal{A}_v$. Hence, $\mathcal{L}_{\downarrow b,0}^x \in \mathcal{A}_v$. We also have that $L_0^x \neq \emptyset$ since $P_x \in L_0^x$. Consequently, each correct $P_x \in H$ verifies that his $\mathcal{L}_{\downarrow b,0}^x \in \mathcal{A}_v$, $L_0^x \neq \emptyset$ and decides on 0 at Step 2.5.
- ($v = 1$) Because P_s is correct, Proxcast_b guarantees that every $P_i \in H$ has $\ell_i = b - 1$. Now consider any correct P_x . We have that $H \subseteq L_{b-1}^x$. Because \mathcal{A}_v is monotone and $\mathcal{P}' \setminus H \in \mathcal{A}_v$, we get that $\mathcal{P}' \setminus L_{b-1}^x \in \mathcal{A}_v$. Hence, $\mathcal{L}_{\downarrow b-1,b}^x \in \mathcal{A}_v$. Assume for the sake of contradiction that P_x decides on 0 instead of 1. This means that $\mathcal{L}_{\downarrow b,0}^x \in \mathcal{A}_v$, $\bigwedge_{k=0}^{b-1} (L_k^x \neq \emptyset)$ and $\bigwedge_{k=0}^{b-2} (\mathcal{L}_{\downarrow k,k+1}^x \in \mathcal{A}_c)$. Together with $\mathcal{L}_{\downarrow b-1,b}^x \in \mathcal{A}_v$ this implies that \mathcal{L}^x is a proper $(b+1)$ -partition of \mathcal{P} , showing that $(\mathcal{A}_c, \mathcal{A}_v)$ is not $(b+1)$ -chain-free-compatible, a contradiction.

CONSISTENCY: If P_s is correct, then consistency holds because of the validity property. Assume now the adversary corrupts some $A \in \mathcal{A}_c$ such that $P_s \in A$. Let $H := \mathcal{P}' \setminus A$ denote the set of correct receivers. Because HybridBC satisfies the consistency property if $A \setminus \{P_s\} \in \mathcal{A}'_c$ is corrupted, we have that receivers in H compute the same set \mathcal{L} , i.e., for all $P_i, P_j \in H$ holds $\mathcal{L}^i = \mathcal{L}^j$.

Consider a party $P_i \in H$ with the smallest ℓ_i . The properties of Proxcast_b guarantee that any $P_j \in H$ has $\ell_j \in \{\ell_i, \ell_i + 1\}$. If $\ell_j = \ell_i$, then P_j decides on the same value as P_i at Step 2.5 because P_j uses the same \mathcal{L} . Assume now that $\ell_j = \ell_i + 1$. Consider two possible cases:

(P_i decides on 0) If P_i decides on 0, then

$$(\mathcal{L}_{\downarrow b,0} \in \mathcal{A}_v) \wedge \left(\bigwedge_{k=0}^{\ell_i} L_k^i \neq \emptyset \right) \wedge \left(\bigwedge_{k=0}^{\ell_i-1} \mathcal{L}_{\downarrow k,k+1} \in \mathcal{A}_c \right)$$

is true. Because any correct $P_x \in H$ has $\ell_x \in \{\ell_i, \ell_i + 1\}$, we have that $H \subseteq L_{\ell_i} \cup L_{\ell_i+1}$. Because \mathcal{A}_c is monotone, we get that $\mathcal{P}' \setminus (L_{\ell_i} \cup L_{\ell_i+1}) \in \mathcal{A}_c$. Hence, $\mathcal{L}_{\downarrow \ell_i, \ell_i+1} \in \mathcal{A}_c$. Since $P_j \in L_{\ell_i+1}$, we have that $L_{\ell_i+1} \neq \emptyset$. Consequently, P_j verifies that

$$(\mathcal{L}_{\downarrow b,0} \in \mathcal{A}_v) \wedge \left(\bigwedge_{k=0}^{\ell_i+1} L_k^i \neq \emptyset \right) \wedge \left(\bigwedge_{k=0}^{\ell_i} \mathcal{L}_{\downarrow k,k+1} \in \mathcal{A}_c \right)$$

is true and decides on 0.

(P_i decides on 1) If P_i decides on 1, then the following formula is false:

$$(\mathcal{L}_{\downarrow b,0} \in \mathcal{A}_v) \wedge \left(\bigwedge_{k=0}^{\ell_i} L_k^i \neq \emptyset \right) \wedge \left(\bigwedge_{k=0}^{\ell_i-1} \mathcal{L}_{\downarrow k,k+1} \in \mathcal{A}_c \right).$$

If the above formula is false, then so is the one below:

$$(\mathcal{L}_{\downarrow b,0} \in \mathcal{A}_v) \wedge \left(\bigwedge_{k=0}^{\ell_i+1} L_k^i \neq \emptyset \right) \wedge \left(\bigwedge_{k=0}^{\ell_i} \mathcal{L}_{\downarrow k,k+1} \in \mathcal{A}_c \right).$$

Hence, P_j also decides on 1. □

4.3 The Broadcast Protocol

In order to achieve broadcast secure against an adversarial structure \mathcal{A} which is $(b + 1)$ -chain-free, we let the parties invoke the hybridly secure broadcast protocol with \mathcal{A}_c and \mathcal{A}_v set to \mathcal{A} .

Protocol Broadcast($\mathcal{P}, P_s, v, \mathcal{A}$)

1. Parties \mathcal{P} invoke **HybridBC**($\mathcal{P}, P_s, v, \mathcal{A}, \mathcal{A}$). Let y_i denote the output each P_i receives.
2. $\forall P_i \in \mathcal{P}$: decide on y_i .

Note that the protocol **Broadcast** achieves broadcast for binary domains only. In order to achieve broadcast for arbitrary input domains efficiently one can use broadcast amplification protocols of [HMR14, HR14].

Lemma 6. *In the b -minicast model, the protocol **Broadcast** perfectly securely achieves broadcast if \mathcal{A} is $(b + 1)$ -chain-free.*

Proof. Due to Lemma 4 we have that if \mathcal{A} is $(b + 1)$ -chain-free, then $(\mathcal{A}, \mathcal{A})$ is $(b + 1)$ -chain-free-compatible. Hence the invocation of the protocol **HybridBC** at Step 1 achieves broadcast, and so does the protocol **Broadcast**. \square

5 The Impossibility Proof

We employ Lemma 2 of [CFF⁺05]:

Lemma 7. *In the b -minicast communication model, broadcast among $b + 1$ parties $\{Q_0, \dots, Q_b\}$ is not achievable if any pair $Q_i, Q_{(i+1) \bmod (b+1)}$ can be honest while the adversary corrupts the remaining parties.*

Now we proceed to the main impossibility statement (based on [CFF⁺05, Theorem 2]).

Lemma 8. *In the b -minicast communication model, there is no secure broadcast protocol among \mathcal{P} that tolerates an adversarial structure \mathcal{A} which is not $(b + 1)$ -chain-free.*

Proof. For the sake of contradiction, assume that there exists a broadcast protocol π tolerating \mathcal{A} which is not $(b + 1)$ -chain-free. Because \mathcal{A} is not $(b + 1)$ -chain-free, there exists a proper $(b + 1)$ -partition $\mathcal{S} = (S_0, \dots, S_b)$ of \mathcal{P} such that all $\mathcal{S}_{\downarrow i, i+1} \in \mathcal{A}$. Using π , we now construct a protocol π' among $b + 1$ parties $\{Q_0, \dots, Q_b\}$ for achieving broadcast from b -minicast. The protocol π' lets each Q_i simulate parties in S_i . If a party Q_i is corrupted, then the simulated parties in S_i can behave arbitrarily. If a party Q_i is honest, then the simulated parties in S_i follow their protocol specification, i.e., behave correctly. Because π is secure against corruption of any set $\mathcal{S}_{\downarrow i, i+1}$, the protocol π' is secure whenever any pair $Q_i, Q_{i+1 \bmod b+1}$ is honest and the remaining parties are corrupted. This contradicts Lemma 7. \square

6 Conclusions

We showed that broadcast secure against any adversarial structure \mathcal{A} is achievable from b -minicast channels if and only if \mathcal{A} is $(b + 1)$ -chain-free. This result is a generalization of [PSL80, FM98, FM00, CFF⁺05, RVS⁺04, CDI⁺13]. An interesting open question is to continue this line of research

and study broadcast achievability in communication models where only some subset of b -minicast channels is available.

Acknowledgments. We would like to thank Martin Hirt, Sandro Coretti and anonymous referees for their valuable comments about the paper.

References

- [BBCK14] M. Backes, F. Bendun, A. Choudhury, and A. Kate. Asynchronous MPC with a strict honest majority using non-equivocation. In M. M. Halldórsson and S. Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 10–19. ACM, 2014.
- [BGP92] P. Berman, J. A. Garay, and K. J. Perry. Bit optimal distributed consensus. In *Computer Science Research*, pages 313–322. Plenum Publishing Corporation, New York, NY, USA, 1992.
- [CDI⁺13] G. Cohen, I. B. Damgård, Y. Ishai, J. Kölker, P. B. Miltersen, R. Raz, and R. D. Rothblum. Efficient multiparty protocols via log-depth threshold formulae - (extended abstract). In R. Canetti and J. A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 185–202. Springer, 2013.
- [CFF⁺05] J. Considine, M. Fitzi, M. Franklin, L. A. Levin, U. Maurer, and D. Metcalf. Byzantine agreement given partial broadcast. *Journal of Cryptology*, 18(3):191–217, July 2005.
- [CW92] B. A. Coan and J. L. Welch. Modular construction of a byzantine agreement protocol with optimal message bit complexity. *Information and Computation*, 97:61–85, March 1992.
- [DS83] D. Dolev and H. R. Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.
- [FHHW03] M. Fitzi, M. Hirt, T. Holenstein, and J. Wullschleger. Two-threshold broadcast and detectable multiparty computation. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 51–67. Springer, 2003.
- [FM98] M. Fitzi and U. M. Maurer. Efficient byzantine agreement secure against general adversaries. In S. Kutten, editor, *DISC*, volume 1499 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 1998.
- [FM00] M. Fitzi and U. Maurer. From partial consistency to global broadcast. In F. Yao, editor, *Proc. 32nd ACM Symposium on Theory of Computing — STOC 2000*, pages 494–503. ACM, May 2000.
- [GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the 19th annual ACM symposium on Theory of computing, STOC '87*, pages 218–229, New York, NY, USA, 1987. ACM.
- [HM97] M. Hirt and U. M. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In J. E. Burns and H. Attiya, editors, *PODC*, pages 25–34. ACM, 1997.
- [HMR14] M. Hirt, U. Maurer, and P. Raykov. Broadcast amplification. In Y. Lindell, editor, *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 419–439. Springer, 2014.
- [HMZ08] M. Hirt, U. M. Maurer, and V. Zikas. MPC vs. SFE : Unconditional and computational security. In J. Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2008.
- [HR14] M. Hirt and P. Raykov. Multi-valued byzantine broadcast: The $t < n$ case. In P. Sarkar and T. Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 448–465. Springer, 2014.
- [HT13] M. Hirt and D. Tschudi. Efficient general-adversary multi-party computation. In K. Sako and P. Sarkar, editors, *Advances in Cryptology — ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 181–200. Springer-Verlag, December 2013.
- [JMS12] A. Jaffe, T. Moscibroda, and S. Sen. On the price of equivocation in byzantine agreement. In D. Kowalski and A. Panconesi, editors, *ACM Symposium on Principles of Distributed Computing, PODC '12, Funchal, Madeira, Portugal, July 16-18, 2012*, pages 309–318. ACM, 2012.

- [LO14] J. Lamkins and R. Ostrovsky. Communication-efficient MPC for general adversary structures. In M. Abdalla and R. D. Prisco, editors, *Security and Cryptography for Networks - 9th International Conference, SCN 2014, Amalfi, Italy, September 3-5, 2014. Proceedings*, volume 8642 of *Lecture Notes in Computer Science*, pages 155–174. Springer, 2014.
- [Mau02] U. M. Maurer. Secure multi-party computation made simple. In S. Cimato, C. Galdi, and G. Persiano, editors, *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, volume 2576 of *Lecture Notes in Computer Science*, pages 14–28. Springer, 2002.
- [PSL80] M. C. Pease, R. E. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.
- [PW96] B. Pfitzmann and M. Waidner. Information-theoretic pseudosignatures and Byzantine agreement for $t \geq n/3$. Technical report, IBM Research, 1996.
- [RVS⁺04] D. V. S. Ravikant, M. Venkitasubramaniam, V. Srikanth, K. Srinathan, and C. P. Rangan. On byzantine agreement over $(2, 3)$ -uniform hypergraphs. In R. Guerraoui, editor, *Distributed Computing, 18th International Conference, DISC 2004, Amsterdam, The Netherlands, October 4-7, 2004, Proceedings*, volume 3274 of *Lecture Notes in Computer Science*, pages 450–464. Springer, 2004.
- [Yao82] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, SFCS '82*, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.