Strong Separations Between Broadcast and Authenticated Channels

Julian Loss
Ruhr University Bochum, Germany
julian.loss@rub.de
https://orcid.org/0000-0002-7979-3810

Ueli Maurer
ETH Zurich, Switzerland
maurer@inf.ethz.ch

Daniel Tschudi[1]
Aarhus University, Denmark
tschudi@cs.au.dk
https://orcid.org/0000-0001-6188-1049

Abstract

In the theory of distributed systems and cryptography one considers a setting with n parties, (often) connected via authenticated bilateral channels, who want to achieve a certain goal even if some fraction of the parties is dishonest. A classical goal of this type is to construct a broadcast channel. A broadcast channel guarantees that all honest recipients get the same value v (consistency) and, if the sender is honest, that v is the sender's input (validity). Lamport et al. showed that it is possible to construct broadcast if and only if the fraction of cheaters is less than a third.

A natural question, first raised by Lamport, is whether there are weaker, still useful primitives achievable from authenticated channels. He proposed weak broadcast, where the validity condition must hold only if all parties are honest, and showed that it can be achieved with an unbounded number of protocol rounds, while broadcast cannot, suggesting that weak broadcast is in a certain sense weaker than broadcast.

The purpose of this paper is to deepen the investigation of the separation between broadcast and authenticated channels. This is achieved by proving the following results. First, we prove a stronger impossibility result for 3-party broadcast. Even if two of the parties can broadcast, one can not achieve broadcast for the third party. Second, we prove a strong separation between authenticated channels and broadcast by exhibiting a new primitive, called XOR-cast, which satisfies two conditions: (1) XOR-cast is strongly unachievable (even with small error probability) from authenticated channels (which is not true for weak broadcast), and (2) broadcast is strongly unachievable from XOR-cast (and authenticated channels). This demonstrates that the hierarchy of primitives has a more complex structure than previously known. Third, we prove a strong separation between weak broadcast and broadcast which is not implied by Lamport's results. The proofs of these results requires the generalization of known techniques for impossibility proofs.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols

Keywords and phrases cryptography, multi-party computation, broadcast, impossibility

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.36

[1] Author was supported by advanced ERC grant MPCPRO.

32nd International Symposium on Distributed Computing (DISC 2018).
Editors: Ulrich Schmid and Josef Widder; Article No. 36; pp. 36:1–36:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Broadcast and Weaker Consistency Guarantees

In the theory of distributed systems and in cryptography one often considers a set of n parties which must securely perform a certain computation, even if some of the parties are dishonest. Broadcast, one of the most fundamental and widely used such primitives, allows one (possibly cheating) party to distribute a value m consistently to the other parties, in a context where only bilateral (authenticated) channels between parties are available. More formally, a broadcast protocol allows a sender to distribute a value v_s such that: *Consistency:* Every honest party outputs the same value v. *Validity:* If the sender is honest, the honest parties output the sender's value $v = v_s$. The seminal result of [12] and [10] states that given authenticated channels, broadcast can be achieved if and only if strictly less than $\frac{n}{3}$ of the involved parties behave dishonestly, even if an error probability of less than $\frac{1}{3}$ were tolerated.

In this work, consistency guarantees of a primitive, e.g. a broadcast channel, to which (potentially) each party has an input and receives an output, are modelled in a very general and natural manner, using so-called consistency specifications [14]. It captures, for every set H of (assumed) honest parties and for every tuple of input values of these honest parties, which tuples of output values are possible, no matter what the other parties do. In other words, a specification guarantees that no adversarial behavior can result in the honest parties' output values to be outside the specified set of tuples. Note that while this concept captures consistency guarantees in the most general form, it does (intentionally) not capture secrecy guarantees.

Broadcast guarantees a very strong form of consistency. The study of primitives with a weaker form of consistency guarantee is well-motivated for two different reasons described below.

First, as argued by Lamport in [11], there are settings of practical relevance where a weaker form of broadcast is sufficient. Specifically, in the *transaction commit problem*, a database transaction is coordinated by some (not necessarily honest) party P_1 who decides whether a transaction should be committed or aborted. A single dishonest party P_i may be enough to cause the transaction to be aborted, but in this case, the honest parties must agree on whether to abort the transaction, or to commit to it. To formalize this setting, [11] introduced a weaker form of broadcast, which we will henceforth refer to as a *weak broadcast channel*. This channel behaves like a regular broadcast channel if all parties are honest, but requires the validity condition to hold *only if every party is honest*. Such a guarantee may be achievable even if broadcast is not achievable.

Second, such a weaker primitive P might be assumed to be available, and one can ask whether a stronger primitive (e.g. a broadcast channel) can be achieved by a protocol that not only can use authenticated channels, but also has access to P. A result of this type, proved in [4], is that broadcast is achievable up to $\frac{n}{2}$ cheaters, assuming that each party can broadcast to any two other parties.

The ultimate goal of a theory in this field is a characterization of various levels of consistency guarantees as well as the hierarchy between them.

1.2 Contribution and Outline

In this work, we are concerned with refining the hierarchy between different types of consistency guarantees and placing weak broadcast in such a hierarchy. As is common for impossibility results in distributed computing, we first prove all of our results in the setting of three parties and then generalize them to the n-party setting.

In order to strengthen the known impossibility result of [12] one can investigate whether it still holds, even if certain primitives are available to the parties, in addition to bilateral authenticated communication. We prove (see Section 4.1) that even if two of the three parties can broadcast values, there is no protocol that would allow the third party to broadcast a value. The proof of this result requires the generalization of known techniques for impossibility proofs to a setting where additional primitives are given. This contribution, which is used throughout the paper, is of independent interest beyond the specific results of this paper.

In order to investigate the hierarchy of consistency primitives between authenticated channels and broadcast, we propose an intermediate level which we call XOR-cast (see Section 4.2). This channel takes a bit b_1 from P_1 and a bit b_2 from P_2 as input. If all parties behave correctly, the value of $b_1 \oplus b_2$ should be output by all parties. If one of the parties P_1 or P_2 is dishonest, the honest parties must output the same value. If P_3 is dishonest, the remaining parties must output $b_1 \oplus b_2$.

We demonstrate a strong separation between authenticated channels and broadcast by proving two strong impossibility results, where we call an impossibility *strong* if it holds even if a constant error probability is tolerated and even if an arbitrary number of communication rounds are allowed. First, it is strongly impossible to achieve XOR-cast from authenticated communication. Second, it is strongly impossible to achieve broadcast from XOR-cast and authenticated communication. This demonstrates that the hierarchy of primitives has a more complex structure than previously known.

The outline of our paper is as follows. In Section 2.1, we introduce the notion of consistency specifications and protocols. We also give some motivating examples of consistency specifications that will be used throughout our work. Here, we extend the work of [14] to case of probabilistic protocols.

In Section 3, we introduce the impossibility proof technique used in this work. In Section 4, we prove our main results, as explained above. Finally, in Section 5 we show how to generalize the results to the n-party case.

1.3 Related Work

Results on the possibility and impossibility of achieving broadcast when other primitives (stronger than authenticated communication) are available were proved in [4, 1, 6, 16, 13]. In a related line of work, [9, 15] derive combinatorial lower bounds on the number of partial broadcast channels among a set of parties needed in order to still be able to achieve broadcast. The general problem of constructing consistency primitives from assumed such primitives was proposed and formalized in [14].

In [11, 2] it is shown that there exists no perfectly secure protocol which constructs weak broadcast from authenticated channels in a finite number of rounds if $\frac{n}{3}$ or more of the parties behave dishonestly. On the other hand, Lamport provides a protocol which achieves weak broadcast, but requires an infinite amount of runtime. This suggests that weak broadcast is in some sense weaker than broadcast; namely, the result in [12] implies that there exists no such approximation protocol for broadcast. However, in distributed computing or MPC one is mostly interested in protocols which run for a fixed number of rounds (or at least terminate eventually). Here, Lamport's results show that both weak broadcast and broadcast cannot be achieved with zero error probability given authenticated channels. If one allows protocols with an error probability negligible in the number of rounds, the impossibility for broadcast still holds. On the other hand, it was shown in [3] that weak broadcast can be achieved from authenticated channels with arbitrary small error probability. Moreover, [12, 11, 3] do not consider the relation between weak broadcast and broadcast. Especially, it is not shown whether broadcast can be achieved given weak broadcast.

Upper bounds for probabilistic broadcast and Byzantine agreement were also studied in [10, 5]. [10] gives an upper bound of $\frac{2}{3}$ (for the success probability) for the fully synchronous, round-based setting. Somewhat surprisingly, [5] consider a synchronous model with a rushing adversary that can observe the inputs of all other parties in each round before deciding on its own input for the round. In this setting, [5] show the stronger bound of $(\sqrt{5}-1)/2$ and also give protocols that match this bound. Such a stronger bound is possible only because the guarantee is stronger and includes a secrecy guarantee: the adversary must not learn the output too early.

2 Preliminaries and Notation

Let $\mathcal{P} = \{P_1, ..., P_n\}$ be a set of n parties (also known as players or processors). For convenience, we will sometimes use i instead of P_i. We distinguish between the subset of honest parties, $H \subseteq \mathcal{P}$, and the dishonest parties in the complement, $\mathcal{P} \setminus H$. Honest parties will execute protocol instructions whereas dishonest parties can deviate arbitrarily from the protocol. For a set M and a subset $S \subseteq \mathcal{P}$, we denote by M^S the Cartesian product $\bigtimes_{i \in S} M$. Moreover we write $[n]$ for the set $\{1, \ldots, n\}$.

2.1 Consistency Specifications

Primitives, such as a broadcast channel, provide the honest parties with consistency guarantees. That is, for every set H of honest parties and every possible choice $\vec{x}_H$ of inputs, the consistency guarantees restrict the set of possible outputs of the honest parties. In this manner, consistency guarantees limit the influence of dishonest parties on the possible outputs of honest parties. We thus model such primitives as functions called *consistency specifications* that map a set of honest parties along with their inputs to a non-empty set of possible outputs. A smaller set of possible outputs implies stronger guarantees offered by the consistency specification, as the uncertainty over the actual output is smaller. More formally, a consistency specification (introduced in [14]) with input domain $\mathcal{D}$ and output domain $\mathcal{R}$ is defined as follows.

▶ **Definition 1.** A *consistency specification* with input domain $\mathcal{D}$ and output domain $\mathcal{R}$ is a function which assigns to every non-empty subset $H \subseteq \mathcal{P}$ and every input tuple $\vec{x}_H \in \mathcal{D}^H$ a non-empty set $\mathcal{C}(H, \vec{x}_H) \subseteq \mathcal{R}^H$ of output tuples and satisfies the following monotonicity constraint: For any non-empty subset $H' \subseteq H \subseteq \mathcal{P}$

$$\mathcal{C}(H, \vec{x}_H)|_{H'} \subseteq \mathcal{C}(H', \vec{x}_{H|H'}). \tag{1}$$

The monotonicity constraint ensures that larger sets of honest parties do not have weaker consistency guarantees. It is therefore natural to require that $\mathcal{C}(H, \vec{x}_H)$ is non-empty for any choice of H and $\vec{x}_H$ as having no output is as good as has having an arbitrary output.

Important Consistency Specifications. We consider two important examples of consistency specifications that we will use throughout this work.

▶ **Definition 2.** A bit *broadcast channel* BC_i for sender P_i can be defined as the following consistency specification

$$\mathrm{BC}_i(H, \vec{x}_H) = \left\{ \vec{y}_H \in \{0,1\}^H \;\middle|\; \begin{array}{l} \exists v \; \big(\; (\forall j \in H : \vec{y}_{H|\{j\}} = v) \\ \wedge \; (i \in H \Rightarrow v = \vec{x}_{H|\{i\}}) \; \big) \end{array} \right\}.$$

The top right line ensures consistency (all honest parties output the same bit) and the bottom right line ensures validity (if the sender is honest, the output bit is its input bit) condition.

▶ **Definition 3.** An *authenticated bit-channel* $\textsc{Auth}_{i,j}$ from P_i to P_j can be defined as the following consistency specification

$$\textsc{Auth}_{i,j}(H, x_H) = \left\{ \vec{y}_H \in \{0,1\}^H \;\middle|\; i,j \in H \Rightarrow \vec{y}_{H|\{j\}} = \vec{x}_{H|\{i\}} \right\}.$$

It guarantees that P_j's output is equal to the input of P_i if both of them are honest.

In the above examples, the inputs of all (honest) parties except P_i have no influence on the consistency guarantee. Similarly for $\textsc{Auth}_{i,j}$, the outputs of all (honest) parties except P_j provide no information (they are arbitrary). We say that such parties have no input, respectively no output. Formally, we define empty inputs and outputs as follows.

▶ **Definition 4.** Let $\mathcal{C}$ be a consistency specification with input domain $\mathcal{D}$ and output domain $\mathcal{R}$. A party P_i has *no input* if for every H with $P_i \in H$ and all $\vec{a}_H, \vec{b}_H \in \mathcal{D}^H$ with $\vec{a}_H|_{H\setminus\{i\}} = \vec{b}_H|_{H\setminus\{i\}}$ it holds that $\mathcal{C}(H, \vec{a}_H) = \mathcal{C}(H, \vec{b}_H)$. A party P_i has *no output* if for every H with $P_i \in H$ and all $\vec{x}_H$ it holds that $\mathcal{C}(H, \vec{x}_H)|_{\{i\}} = \mathcal{R}$.

Finally, we note that the parallel composition of several consistency specifications once again forms a consistency specification. More formally, consider consistency specifications $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\ell)}$ where $\mathcal{C}^{(j)}$ has input domain $\mathcal{D}_j$ and output domain $\mathcal{R}_j$ $j \in [\ell]$. The parallel composition of $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\ell)}$ is defined as follows.

▶ **Definition 5.** The *parallel composition of* $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\ell)}$ is the $(\mathcal{D}, \mathcal{R})$-consistency specification $[\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\ell)}]$ where $\mathcal{D} = \prod_{j\in[\ell]} \mathcal{D}_j$, $\mathcal{R} = \prod_{j\in[\ell]} \mathcal{R}_j$, and for every $H \subseteq \mathcal{P}$ and all $\vec{x}_H = \big((x_{ij})_{j\in[\ell]}\big)_{i\in H} \in \mathcal{D}$ it holds that

$$[\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\ell)}](H, \vec{x}_H) = \left\{ \vec{y}_H \in \mathcal{R} \;\middle|\; \begin{array}{l} \vec{y}_H = \big((y_{ij})_{j\in[\ell]}\big)_{i\in H} \\ \wedge\ \forall j\ (y_{ij})_{i\in H} \in \mathcal{C}^{(j)}\big(H, (x_{ij})_{i\in H}\big) \end{array} \right\}.$$

The complete network of authenticated channels can be seen as the parallel composition of authenticated channels.

▶ **Definition 6.** The complete network $\textsc{Auth}$ of authenticated bit-channels for parties $\mathcal{P}$ is the parallel composition of the set $\{\textsc{Auth}_{i,j} \mid P_i, P_j \in \mathcal{P}\}$ of all authenticated bit-channels.

2.2 Protocols and Constructions.

Protocols are means to construct new consistency specifications from given consistency specifications. A protocol execution is round-based and proceeds as follows. In each round, a party computes an input to the consistency specification used in this round. This input may depend on its protocol input and outputs from previously invoked consistency specifications. At the end of the protocol execution, each party computes its protocol output as a function of its protocol input and all the outputs it received from invoked specifications over the course of the protocol.

Deterministic Protocols. A deterministic protocols runs for $\ell \geq 0$ rounds. In each round r, party P_i uses the deterministic round function $f_i^{(r)}$ to compute its input for the round specification $\mathcal{C}^{(r)}$ which has input domain $\mathcal{D}_r$ and output domain $\mathcal{R}_r$. At the end of the last round, party P_i uses its output function g_i to compute its protocol output. Denote by $\vec{\mathcal{C}} = (\mathcal{C}^{(r)})_{r\in 1,\ldots,\ell}$ the tuple of invoked specifications. Then we can define a deterministic protocol as follows.

▶ **Definition 7** ([14]). A deterministic *ℓ-round protocol* π for tuple $\vec{\mathcal{C}}$ with input domains $\mathcal{D}$ and output domains $\mathcal{R}$ consists of round functions

$$f_i^{(r)} : \mathcal{D} \times \mathcal{R}_1 \times \cdots \times \mathcal{R}_{(r-1)} \to \mathcal{D}_r \quad \forall i \in \mathcal{P}\ \forall r \in [\ell]$$

and output functions

$$g_i : \mathcal{D} \times \mathcal{R}_1 \times \cdots \times \mathcal{R}_{(\ell)} \to \mathcal{R} \quad \forall i \in \mathcal{P}.$$

We explicitly allow zero-round protocols where no consistency specifications are invoked. By executing the protocol π using tuple $\vec{\mathcal{C}}$, the parties achieve a new consistency specification denoted by $\pi\vec{\mathcal{C}}$. The following definition formally defines how the output of $\pi\vec{\mathcal{C}}$ is computed by iteratively applying the round functions of π to the input tuple $\vec{x}_H$.

▶ **Definition 8.** For a protocol π and the corresponding tuple $\vec{\mathcal{C}}$ the *protocol specification* $\pi\vec{\mathcal{C}}$ is the following consistency specification, such that for every $H \subseteq \mathcal{P}$ and $\vec{x}_H = (x_i)_{i\in H} \in \mathcal{D}^H$, we have:

$$\pi\vec{\mathcal{C}}(H, \vec{x}_H) = \left\{ (y_i)_{i\in H} \in \mathcal{R}^H \;\middle|\; \begin{array}{l} \forall r \in [\ell]\ \exists (x_{ir})_{i\in H} \in \mathcal{D}_r^H\ \exists (y_{ir})_{i\in H} \in \mathcal{R}_r^H \\ \forall i \in H\text{:}\ \ x_{ir} = f_i^{(r)}(x_i, y_{i1}, \ldots, y_{ir-1}) \\ \ \wedge\ (y_{ir})_{i\in H} \in \mathcal{C}^{(r)}\big(H, (x_{ir})_{i\in H}\big) \\ \wedge\ \forall i \in H\ y_i = g_i(x_i, y_{i1}, \ldots, y_{i\ell}) \end{array} \right\}.$$

The goal of a protocol execution is to achieve a consistency specification whose guarantees are at least as strong as the guarantees of some target specification $\mathcal{C}$. As already argued, the consistency guarantee becomes stronger as the set of possible outputs becomes smaller. Therefore, we say that a protocol π *constructs* a consistency specification $\mathcal{C}$ from the tuple $\vec{\mathcal{C}}$, if the set of possible outputs of the protocol specification $\pi\vec{\mathcal{C}}(H, \vec{x}_H)$ for arbitrary inputs $H, \vec{x}_H$ to $\pi\vec{\mathcal{C}}$ is a subset of the corresponding set of possible outputs $\mathcal{C}(H, \vec{x}_H)$ of the target specification $\mathcal{C}$. Formally:

▶ **Definition 9.** A protocol π *constructs* a specification $\mathcal{C}$ from the tuple $\vec{\mathcal{C}}$ if we have for all $H \subseteq \mathcal{P}$ and all $\vec{x}_H$ $\pi\vec{\mathcal{C}}(H, \vec{x}_H)$ that $\subseteq \mathcal{C}(H, \vec{x}_H)$.

Often, one is interested in a broader notion of construction where specifications from a set $\mathfrak{C}$ may be invoked arbitrarily often during a protocol execution.

▶ **Definition 10.** A specification $\mathcal{C}$ can be constructed from a set of specifications $\mathfrak{C}$, denoted by $\mathfrak{C} \longrightarrow \mathcal{C}$, if there exists a tuple $\vec{\mathcal{C}}$ of specifications from $\mathfrak{C}$ (including parallel compositions) which allows to construct $\mathcal{C}$.

The above definition naturally extends to a construction notion among sets of consistency specifications: A set of consistency specifications $\mathfrak{C}'$ is constructible from $\mathfrak{C}$, denoted by $\mathfrak{C} \longrightarrow \mathfrak{C}'$ if all $\mathcal{C} \in \mathfrak{C}'$ can be constructed from $\mathfrak{C}$.

Probabilistic Protocols. In a probabilistic protocol, the parties may additionally use local randomness during the protocol execution. Formally, probabilistic protocols are modeled as distributions over deterministic protocols.

▶ **Definition 11.** A *probabilistic protocol* ℓ-round $\mathbf{\Pi}$ for tuple $\vec{\mathcal{C}}_{\mathbf{\Pi}}$ with input domains $\mathcal{D}$ and output domains $\mathcal{R}$ is a random variable (for some distribution) over a set of deterministic protocols of at most ℓ-rounds for tuple $\vec{\mathcal{C}}_{\mathbf{\Pi}}$ with input domains $\mathcal{D}$ and output domains $\mathcal{R}$.

Note that our definition allows for protocols where parties have access to correlated randomness. We denote by $\mathbf{\Pi}\vec{\mathcal{C}}_{\mathbf{\Pi}}$ the random variable over the protocol specifications for $\mathbf{\Pi}$ and $\vec{\mathcal{C}}_{\mathbf{\Pi}}$. A protocol constructs a target specification $\mathcal{C}$ within ϵ if with probability strictly larger than $1-\epsilon$, $\mathbf{\Pi}\vec{\mathcal{C}}_{\mathbf{\Pi}}$ provides better consistency guarantees than $\mathcal{C}$. Formally:

▶ **Definition 12.** A probabilistic protocol $\mathbf{\Pi}$ for tuple $\vec{\mathcal{C}}_{\mathbf{\Pi}}$ constructs $\mathcal{C}$ within ϵ if

$$\min_{H,\vec{x}_H} \mathsf{P}\big(\mathbf{\Pi}\vec{\mathcal{C}}_{\mathbf{\Pi}}(H,\vec{x}_H) \subseteq \mathcal{C}(H,\vec{x}_H)\big) > 1-\epsilon.$$

A construction is called *perfect* if $\epsilon = 0$. A specification $\mathcal{C}$ can be constructed within ϵ from a set $\mathfrak{C}$, denoted by $\mathfrak{C} \xrightarrow{\epsilon} \mathcal{C}$, if there exists a tuple $\vec{\mathcal{C}}_{\mathbf{\Pi}}$ from $\mathfrak{C}$ which allows to construct $\mathcal{C}$ within ϵ.

Note that any deterministic construction is a perfect construction.

3 Impossibility Proofs

In this section, we consider a generalized version of so called 'scenario'-proofs (see e.g., [2]). This proof technique, a special type of proof by contradiction, is normally used to prove that a specification, e.g., broadcast, cannot be constructed from authenticated channels within some ϵ. Here, we extend 'scenario'-proofs to the setting where parties are given additional setup. This means that we want to prove statements of the form "There is no construction of a specification $\mathcal{C}$ from given specifications $\mathfrak{C}$ within ϵ" where $\mathfrak{C}$ is arbitrary set of specification which contains the complete network of authenticated channels.

More formally, the technique allows to prove a *claim* of the form: "$\mathcal{C}$ cannot be constructed from $\mathfrak{C}$ within $\epsilon = \frac{1}{k}$ where $\textsc{Auth} \in \mathfrak{C}$." The corresponding *'scenario'-proof* goes as follows (for a simple example of such a proof, see the proof of Lemma 14). Towards a contradiction, assume that there exists a protocol $\mathbf{\Pi}$ which allows to construct $\mathcal{C}$ from $\mathfrak{C}$ within $\frac{1}{k}$. This implies that for each party P_i and for each input x_i, there exists a corresponding (probabilistic) protocol system $\Pi_i^{x_i}$ which executes the protocol part of P_i for input x_i[2]. For every other party P_j, the protocol system of a party P_i has an interface where one can connect it to P_j's protocol system. This models the assumption that parties are pair-wise connected via authenticated channels. If the parties are given additional specifications in $\mathfrak{C}$ (e.g., broadcast channels for some parties) or some setup (e.g. shared randomness) during the protocol execution, this is modeled via a system R that provides the functionality of these specifications. In this case, all protocol systems have an additional interface where they expect to be connected to R.

The creative part of the proof is to build a *configuration* of connected protocol systems and R, which has impossible output guarantees. This implies that there is no construction of $\mathcal{C}$ from $\mathfrak{C}$ within $\frac{1}{k}$. More formally, we consider a configuration S and the output vector of selected protocol systems which we denote by the random variable $\mathbf{Y}$. To show that S has impossible output guarantees, we use the following technical lemma.

▶ **Lemma 13.** *Let $A_1,\dots,A_k$ be sets with non-empty union $A = \bigcup_{i=1}^k A_i$ and let $\mathbf{Y}$ be a random variable over some set $U \supseteq A$ such that $\mathsf{P}(\mathbf{Y} \in \bigcap_{i=1}^k A_i) = 0$. Then $\min_i \mathsf{P}(\mathbf{Y} \in A_i) \leq 1-\frac{1}{k}$.*

[2] Such a system can be instantiated, for example, as an interactive Turing machine.

Proof. For convenience we denote for any set B by $\mathsf{P}(B)$ the probability $\mathsf{P}(\mathbf{Y} \in B)$. We denote by $\overline{B}$ the complement of B in U. Using elementary set operations and the union bound we get

$$\begin{aligned}\mathsf{P}(\bigcap_{i=1}^{k} A_i) &= 1 - \mathsf{P}(\bigcup_{i=1}^{k} \overline{A_i}) \geq 1 - \sum_{i=1}^{k} \mathsf{P}(\overline{A_i}) \\ &= 1 - \sum_{i=1}^{k}(1 - \mathsf{P}(A_i)) = 1 - k + \sum_{i=1}^{k} \mathsf{P}(A_i).\end{aligned}$$

As the minimum overall $\mathsf{P}(\mathbf{Y} \in A_i)$ is smaller than the average we finally get

$$\begin{aligned}\min_i \mathsf{P}(\mathbf{Y} \in A_i) &\leq \frac{1}{k}\sum_{i=1}^{k} \mathsf{P}(A_i) \\ &\leq \frac{1}{k}(k - 1 + \mathsf{P}(\bigcap_{i=1}^{k} A_i)) = 1 - \frac{1}{k}.\end{aligned}$$ ◀

To get to a contradiction, we thus need to show that there are k sets (of outputs) $A_1, \ldots, A_k$ with empty intersection, where $\mathbf{Y} \in A_i$ with probability strictly greater than $1 - \frac{1}{k}$ for any i. To do so, we use k so-called *scenarios*. Each scenario describes S as a protocol execution among three parties where exactly one of them is dishonest. With the exception of two systems (for the two honest parties), all parts of S are considered to be the 'attack strategy' of the dishonest party. The initial assumption implies that the outputs of the two honest parties in this scenario must satisfy some consistency guarantee with probability strictly greater than $1 - \frac{1}{k}$. This directly translates into a condition on $\mathbf{Y}$. Namely, for the ith scenario, there must exist a set of outputs A_i such that $\mathsf{P}(\mathbf{Y} \in A_i) > 1 - \frac{1}{k}$. To arrive at the desired contradiction, the k scenarios are chosen such that the intersection of all A_i's is empty and therefore $\mathsf{P}(\mathbf{Y} \in \bigcap_{i=1}^{k} A_i) = 0$. In this case, the above lemma implies that for at least one A_i, it must hold that $\mathsf{P}(\mathbf{Y} \in A_i) \leq 1 - \frac{1}{k}$, thus contradicting the fact that for all i, $\mathsf{P}(\mathbf{Y} \in A_i) > 1 - \frac{1}{k}$ (as required by the assumption of a construction within $\epsilon = \frac{1}{k}$).

4 Results

In this section we consider specifications for party set $\mathcal{P} = \{P_1, P_2, P_3\}$ where all inputs and outputs are bit-strings.

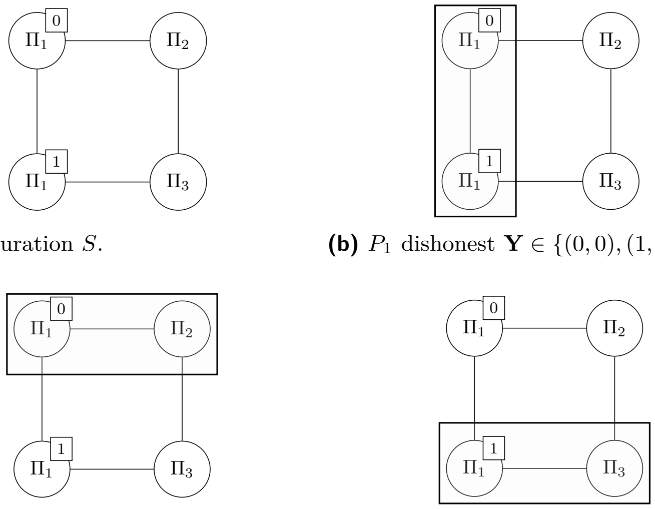
4.1 Strong Broadcast Impossibility

Here, we prove a strong impossibility for the construction of broadcast. That is, we show that broadcast channel, e.g. BC_1, cannot be constructed within $\frac{1}{3}$ even if all other broadcast channels are available. This implies the-well known result by Karlin and Yao [10] that broadcast cannot be constructed from authenticated channels within $\frac{1}{3}$.

As a warm up, we prove first the [10] statement using the impossibility techniques from above.

▶ **Lemma 14.** *[10]* $\textsc{Auth} \overset{\frac{1}{3}}{\not\rightarrow} \mathrm{BC}_1$.

Proof. Towards a contradiction, let us assume that there exists a protocol $\mathbf{\Pi}$ such that $\textsc{Auth} \xrightarrow{\mathbf{\Pi}, \frac{1}{3}} \mathrm{BC}_1$. Then there exist protocol systems $\Pi_1^0, \Pi_1^1, \Pi_2, \Pi_3$. Note that only the



(a) Configuration S.

(b) P_1 dishonest $\mathbf{Y} \in \{(0,0),(1,1)\}$.

(c) P_2 dishonest $\mathbf{Y} \in \{(0,1),(1,1)\}$.

(d) P_3 dishonest $\mathbf{Y} \in \{(0,0),(0,1)\}$.

Figure 1 The configuration S and the three scenarios.

system of P_1 has an input. Each of these systems has two interfaces where it expects to be connected to the systems of the other two parties.

We consider the configuration S in Figure 1a where all four systems are arranged in a circle. The random variable $\mathbf{Y}$ describes the output behavior of systems Π_2 and Π_3. This means that $\mathbf{Y}$ maps to bit-tuples where the first component represents the output of Π_2.

We examine the distribution of $\mathbf{Y}$ using different protocol execution scenarios. First, we consider the scenario where P_2 and P_3 are honest while P_1 is dishonest, i.e., $H = \{P_2, P_3\}$. In this scenario, consistency of broadcast ensures that the outputs of P_2 and P_3 are with probability strictly larger than $1 - \frac{1}{3}$ the same (independently of the behavior of P_1). In the configuration S, this corresponds to the scenario where the system of P_1 consists of the two left-most systems (cf. Figure 1b). This implies that $\mathbf{Y}$ is in $A_1 = \{(0,0),(1,1)\}$ with probability strictly larger than $1 - \frac{1}{3}$. Next, we consider the scenario where P_1 and P_3 are honest ($H = \{P_1, P_3\}$) and P_1 has input 1. In our configuration S, we can perceive the two systems on the top as the system of the dishonest P_2 (cf. Figure 1c). This implies (validity of broadcast) that $\mathsf{P}(\mathbf{Y} \in A_2) > 1 - \frac{1}{3}$ for $A_2 = \{(0,1),(1,1)\}$. Finally, we consider the case $H = \{P_1, P_3\}$ where P_1 has input 0. In our configuration S, we can perceive the two systems at the bottom as the system of the dishonest P_3 (cf. Figure 1d). This implies (validity of broadcast) that $\mathsf{P}(\mathbf{Y} \in A_3) > 1 - \frac{1}{3}$ for $A_3 = \{(0,0),(0,1)\}$.

We observe that $A_1 \cap A_2 \cap A_3 = \varnothing$ and thus $\mathsf{P}(\mathbf{Y} \in \bigcap_{i=1}^{3} A_i) = 0$. This implies with Lemma 13 that for at least one A_i, $\mathsf{P}(\mathbf{Y} \in A_i) \leq 1 - \frac{1}{3}$. This is a contradiction to the fact that $\mathsf{P}(\mathbf{Y} \in A_i) > 1 - \frac{1}{3}$ for all A_i, as required by the definition of a construction within $\epsilon = \frac{1}{3}$. Thus, there exists no ϵ-construction of broadcast for $\epsilon \leq \frac{1}{3}$. ◀

▶ **Theorem 15.** $\{\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3\} \overset{\frac{1}{3}}{\not\to} \mathrm{BC}_1$.

Proof. To prove this result we use the 'scenario'-proof technique from Section 3. Assume therefore that there exists a probabilistic protocol $\mathbf{\Pi}$ which allows to construct BC_1 from $\{\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3\}$ within $\epsilon = \frac{1}{3}$. Thus, there exist protocol systems $\Pi_1^0, \Pi_1^1, \Pi_2, \Pi_3$ where the bit on top of Π_1 denotes the input of sender P_1. Additionally there exists a system $[\mathrm{BC}_2, \mathrm{BC}_3]$ which corresponds to the given broadcast channels for P_2 and P_3.

We first show how to construct a system $\overline{\mathrm{BC}}$ from the system $[\mathrm{BC}_2, \mathrm{BC}_3]$. This system $\overline{\mathrm{BC}}$ will be used to build the configuration S, rather than $[\mathrm{BC}_2, \mathrm{BC}_3]$ directly. Thus, $\overline{\mathrm{BC}}$ corresponds to the system R in our informal description from Section 3. System $\overline{\mathrm{BC}}$ is essentially the same as $[\mathrm{BC}_2, \mathrm{BC}_3]$ except that the interface of P_1 is cloned. More precisely, $\overline{\mathrm{BC}}$ has four interfaces. The two interfaces for parties P_2 and P_3 have the same input/output behavior as in $[\mathrm{BC}_2, \mathrm{BC}_3]$. However, the interface for P_1 appears twice in $\overline{\mathrm{BC}}$, where both copies deliver the same output. Note that this completely describes the behaviour of $\overline{\mathrm{BC}}$, since P_1's interface does not take input in $[\mathrm{BC}_2, \mathrm{BC}_3]$ (and thus, it also does not take an input in $\overline{\mathrm{BC}}$).

System $\overline{\mathrm{BC}}$ can be built from $[\mathrm{BC}_2, \mathrm{BC}_3]$ in three different ways. First, one can build it by adding a system e_1 to the P_1-interface of $[\mathrm{BC}_2, \mathrm{BC}_3]$ which relays the outputs of this interface to the two P_1-interfaces of $\overline{\mathrm{BC}}$. Second, one can build $\overline{\mathrm{BC}}$ from $[\mathrm{BC}_2, \mathrm{BC}_3]$ by adding a system e_2 to the P_2-interface of $[\mathrm{BC}_2, \mathrm{BC}_3]$. System e_2 relays any input at the $\overline{\mathrm{BC}}$ P_2-interface to $[\mathrm{BC}_2, \mathrm{BC}_3]$. Any output at the P_2-interface of $[\mathrm{BC}_2, \mathrm{BC}_3]$ is relayed to the $\overline{\mathrm{BC}}$ P_1-interface and the $\overline{\mathrm{BC}}$ P_2-interfaces of e_2, respectively. Note that adding system e_2 in this way achieves the same as adding e_1. This is true, because in $[\mathrm{BC}_2, \mathrm{BC}_3]$, the outputs at any interface are always identical, due to the consistency guarantees of BC_2 and BC_3. Analogously, one can build $\overline{\mathrm{BC}}$ from $[\mathrm{BC}_2, \mathrm{BC}_3]$ by adding a system e_3 to the P_3-interface of $[\mathrm{BC}_2, \mathrm{BC}_3]$. In summary we have that the systems $\overline{\mathrm{BC}}, e_1[\mathrm{BC}_2, \mathrm{BC}_3], e_2[\mathrm{BC}_2, \mathrm{BC}_3]$, and $e_3[\mathrm{BC}_2, \mathrm{BC}_3]$ have the same input/output behavior.

We consider now the configuration S in Figure 2a and the output $\mathbf{Y}$ of systems Π_2 and Π_3. It follows from the above argumentation that the configurations seen in Figures 2b-2d have the same output behavior $\mathbf{Y}$ as S.

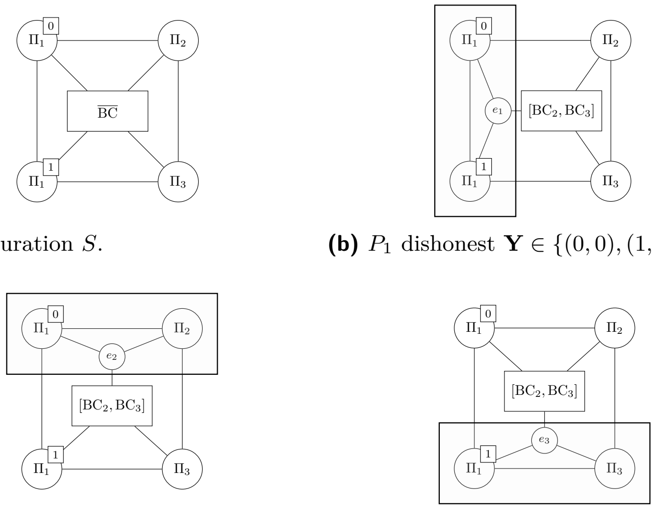
We examine the distribution of $\mathbf{Y}$ using different protocol execution scenarios. First, we consider the scenario where P_1 is dishonest, i.e, $H = \{P_2, P_3\}$. The consistency of BC_1 implies that with probability strictly larger than $1 - \frac{1}{3}$, the outputs of P_2 and P_3 are the same. In this scenario, the adversarial P_1 could control a system consisting of the three left-most systems in Figure 2b. The consistency of broadcast thus implies for S that $\mathsf{P}(\mathbf{Y} \in A_1) > 1 - \frac{1}{3}$, where $A_1 = \{(0,0),(1,1)\}$. Next, we consider the scenario $H = \{P_1, P_3\}$ where P_1 has input 1. Here, dishonest P_2 could run the top-three systems in Figure 2c. The validity condition of BC_1 implies that $\mathsf{P}(\mathbf{Y} \in A_2) > 1 - \frac{1}{3}$ for $A_2 = \{(0,1),(1,1)\}$. Finally, we consider the scenario $H = \{P_1, P_2\}$ where P_1 has input 0. Here, dishonest P_3 could run the bottom-three systems in Figure 2d. The validity condition of BC_1 implies that $\mathsf{P}(\mathbf{Y} \in A_3) > 1 - \frac{1}{3}$ for $A_3 = \{(0,0),(0,1)\}$. The intersection $A_1 \cap A_2 \cap A_3$ is empty and hence $\mathsf{P}(\mathbf{Y} \in A_1 \cap A_2 \cap A_3) = 0$. Now, Lemma 13 implies that for at least one A_i, $\mathsf{P}(\mathbf{Y} \in A_i) \leq 1 - \frac{1}{3}$. This is a contradiction to the fact that $\mathsf{P}(\mathbf{Y} \in A_i) > 1 - \frac{1}{3}$ for all A_i as required by the definition of a construction within $\epsilon = \frac{1}{3}$. Thus no construction of broadcast BC_1 from $\{\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3\}$ exists within $\epsilon = \frac{1}{3}$. ◀

▶ **Corollary 16.** *In particular, for every protocol* $\mathbf{\Pi}$ *which constructs broadcast* BC_1 *from* $\{\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3\}$*, there exists* $H \subseteq \mathcal{P}$ *of size two such that*

$$\mathsf{P}\big(\mathbf{\Pi}(\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3)(H, \vec{x}_H) \subseteq \mathrm{BC}_1(H, \vec{x}_H)\big) \leq 1 - \frac{1}{3}.$$

4.2 Strong Separation of Broadcast and Authenticated Channels

In this section, we prove a strong separation between broadcast and authenticated channels. That is, we present a specification, called XOR-cast, which neither can be constructed from authenticated channels within a constant ϵ, nor is sufficient to construct broadcast within



(a) Configuration S.

(b) P_1 dishonest $\mathbf{Y} \in \{(0,0),(1,1)\}$.

(c) P_2 dishonest $\mathbf{Y} \in \{(0,1),(1,1)\}$.

(d) P_3 dishonest $\mathbf{Y} \in \{(0,0),(0,1)\}$.

Figure 2 The configuration S and the three scenarios. A line between two systems means that they are connected. In the case of protocol systems this corresponds to the fact that parties can communicate over authenticated channels.

a constant ϵ. XOR-cast takes a bit b_i from P_i and a bit b_j from P_j as input. If all parties behave correctly, the value of $b_i \oplus b_j$ should be output by all parties. If one of the parties P_i, P_j is dishonest, the honest parties should output the same value. If the third party P_k is dishonest, the remaining parties should output $b_i \oplus b_j$.

▶ **Definition 17.** Let $P_i, P_j \in \mathcal{P}$ be distinct parties. The *XOR-cast* $\mathrm{XC}_{i,j}$ for P_i and P_j is defined as follows.

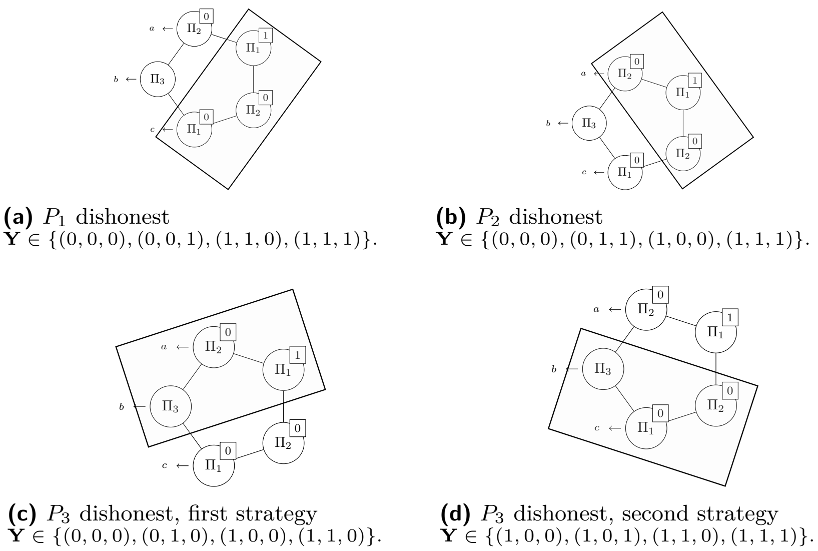
$$\begin{aligned}&\mathrm{XC}_{i,j}(H, \vec{x}_H)\\ &= \left\{ \vec{y} \in \{0,1\}^H \;\middle|\; \begin{array}{l} \exists v \left((\forall \ell \in H : \vec{y}_{H|\{\ell\}} = v) \right. \\ \left. \wedge\ (i,j \in H \Rightarrow v = \vec{x}_{H|\{i\}} \oplus \vec{x}_{H|\{j\}}) \right) \end{array} \right\}.\end{aligned}$$

The top right line in the equation ensures that all honest parties output the same value. The bottom right line ensures for honest P_i and P_j that the output is the XOR of their input-bits.

We first prove that XOR-cast, e.g., $\mathrm{XC}_{1,2}$, cannot be constructed from the network of authenticated channels.

▶ **Lemma 18.** $\{\textsc{Auth}\} \overset{\frac{1}{4}}{\not\rightarrow} \mathrm{XC}_{1,2}$.

Proof. We again use the 'scenario'-proof technique. Towards a contradiction, assume that there exists a protocol allowing to construct $\mathrm{XC}_{1,2}$ from $\{\textsc{Auth}\}$ within $\frac{1}{4}$. Then there exist protocol systems $\Pi_1^{x_1}, \Pi_2^{x_2}, \Pi_3$ for parties P_1, P_2, P_3 where x_1 denotes the input bit of P_1 and x_2 denotes the input bit of P_2. Consider the pentagon configuration S in Figure 3. Let $\mathbf{Y}$ be the random variable over the output (a,b,c) of the three left-most systems, i.e., where a is the output of Π_2^0 (top left), b the output of Π_3 (middle left), and c the output of Π_1^0 (bottom left). We examine the distribution of $\mathbf{Y}$ using four different protocol execution scenarios. First, we consider the scenario where P_2 and P_3 are honest ($H = \{P_2, P_3\}$) and P_2 has input 0. In this scenario, the dishonest P_1 could run the three systems in the bottom-left in Figure 3a. The outputs of P_2 and P_3 must be the same. This implies $\mathsf{P}(\mathbf{Y} \in A_1) > 1 - \frac{1}{4}$



(a) P_1 dishonest
$\mathbf{Y} \in \{(0,0,0),(0,0,1),(1,1,0),(1,1,1)\}$.

(b) P_2 dishonest
$\mathbf{Y} \in \{(0,0,0),(0,1,1),(1,0,0),(1,1,1)\}$.

(c) P_3 dishonest, first strategy
$\mathbf{Y} \in \{(0,0,0),(0,1,0),(1,0,0),(1,1,0)\}$.

(d) P_3 dishonest, second strategy
$\mathbf{Y} \in \{(1,0,0),(1,0,1),(1,1,0),(1,1,1)\}$.

Figure 3 The configuration S and the four scenarios.

for $A_1 = \{(0,0,0),(0,0,1),(1,1,0),(1,1,1)\}$. Next, we consider the scenario $H = \{P_1, P_3\}$ where P_1 has input 0 (cf. Figure 3b). Here, the outputs of P_1 and P_3 must be the same. This implies that $\mathsf{P}(\mathbf{Y} \in A_2) > 1 - \frac{1}{4}$ for $A_2 = \{(0,0,0),(0,1,1),(1,0,0),(1,1,1)\}$. Next, we consider the scenario $H = \{P_1, P_2\}$ where both P_1 and P_2 have input 0 (cf. Figure 3c). Here, the output of P_1 must be $0 = 0 \oplus 0$. This implies that $\mathsf{P}(\mathbf{Y} \in A_3) > 1 - \frac{1}{4}$ for $A_3 = \{(0,0,0),(0,1,0),(1,0,0),(1,1,0)\}$. Finally, we consider the scenario $H = \{P_1, P_2\}$ where P_1 has input 1 and P_2 has input 0 (cf. Figure 3d). Here, the output of P_2 must be $1 = 1 \oplus 0$. This implies that $\mathsf{P}(\mathbf{Y} \in A_4) > 1 - \frac{1}{4}$ for $A_4 = \{(1,0,0),(1,0,1),(1,1,0),(1,1,1)\}$.

We observe that the intersection $A_1 \cap A_2 \cap A_3 \cap A_4$ is empty and hence $\mathsf{P}(\mathbf{Y} \in \bigcap_{i=1}^{4} A_i) = 0$. This implies with Lemma 13 that for at least one A_i, $\mathsf{P}(\mathbf{Y} \in A_i) \leq 1 - \frac{1}{4}$. This is a contradiction to the fact that $\mathsf{P}(\mathbf{Y} \in A_i) > 1 - \frac{1}{4}$ for all A_i as required by the definition of a construction within $\epsilon = \frac{1}{4}$. Thus no construction of $\mathrm{XC}_{1,2}$ from $\textsc{Auth}$ exists within $\frac{1}{4}$. ◀

▶ **Corollary 19.** *In particular, for every protocol $\mathbf{\Pi}$, there exists $H \subset \mathcal{P}, |H| = 2$, such that*

$$\mathsf{P}\big(\mathbf{\Pi}(\textsc{Auth})(H, \vec{x}_H) \subseteq \mathrm{XC}_{1,2}(H, \vec{x}_H)\big) \leq 1 - \frac{1}{4}.$$

Next, we show that one can perfectly construct $\mathrm{XC}_{i,j}$ given the complete network of authenticated channels and a broadcast channel for P_i or P_j.

▶ **Lemma 20.** *For all $i \neq j \in \{1,2,3\}$ $\{\textsc{Auth}, \mathrm{BC}_i\} \longrightarrow \mathrm{XC}_{i,j}$.*

Proof. Let b_i be the input of P_i and let b_j be the input of P_j and denote by P_k the third party. Consider the following protocol.

1. P_j sends b_j to P_i. Denote by $\hat{b}_j$ the bit received by P_i.
2. P_i broadcasts $b_k := b_i \oplus \hat{b}_j$ using BC_i. Denote by $\hat{b}_k$ the bit received by P_j and P_k.
3. P_i outputs b_k, P_j and P_k both output $\hat{b}_k$.

If at least P_i and P_j are honest we have $\hat{b}_j = b_j$ and $\hat{b}_k = b_k$. All honest parties will output $b_k = b_i \oplus b_j$ as required by $\mathrm{XC}_{i,j}$. On the other hand if $H = \{P_j, P_k\}$ both honest parties will output $\hat{b}_k$ as required by $\mathrm{XC}_{i,j}$. If $H = \{P_i, P_k\}$ we have $\hat{b}_k = b_k$. Both honest parties will output b_k as required by $\mathrm{XC}_{i,j}$. If at most one party is honest any output is fine, thus the protocol achieves the construction also in those cases. ◀

$$\begin{tikzcd}
\textsc{Auth} \arrow[r, "\frac{1}{4}", "/" marking] & \mathrm{XC} \arrow[l, bend left, "\epsilon \geq 0."] \arrow[r, "\frac{1}{3}", "/" marking] & \mathrm{BC} \arrow[l, bend left, "\epsilon \geq 0"]
\end{tikzcd}$$

Figure 4 XOR-cast strongly separates AUTH and BC.

Finally, we show that XOR-cast is strictly weaker than broadcast. Even given all three XOR-casts, one cannot construct a single broadcast channel. Without loss of generality, we show that one cannot construct BC_1 given all XOR-casts within $\epsilon \leq \frac{1}{3}$.

▶ **Lemma 21.** $\{\mathrm{XC}_{1,2}, \mathrm{XC}_{1,3}, \mathrm{XC}_{2,3}, \textsc{Auth}\} \overset{\frac{1}{3}}{\not\longrightarrow} \mathrm{BC}_1$.

Proof. Towards a contradiction, let us assume that one can construct BC_1 given the XOR-casts, i.e., $\{\mathrm{XC}_{1,2}, \mathrm{XC}_{1,3}, \mathrm{XC}_{2,3}, \textsc{Auth}\} \longrightarrow \mathrm{BC}_1$ within $\epsilon \leq \frac{1}{3}$. Lemma 20 implies that one can perfectly construct all XOR-casts given broadcast channels $\mathrm{BC}_2, \mathrm{BC}_3$. This implies that one can construct BC_1 from $\{\mathrm{BC}_2, \mathrm{BC}_3, \textsc{Auth}\}$ within $\epsilon \leq \frac{1}{3}$, a contradiction to Lemma 15. ◀

The above lemmas directly imply the following theorem.

▶ **Theorem 22.** *Authenticated channels and broadcast are strongly separated by XOR-cast.*

4.3 Weak Broadcast

For comparison, we consider *weak broadcast* which was introduced in [11]. This specification provides the same consistency guarantees as broadcast except that validity only holds if all parties are honest.

▶ **Definition 23.** Let $P_s \in \mathcal{P}$. A *weak broadcast-channel* wBC_s for sender P_s is defined to be a $(\{0,1\}, \{0,1\})$-consistency specification where for every $H \subseteq \mathcal{P}$ and all $\vec{x}_H \in \{0,1\}^H$ it holds that

$$\begin{aligned}&\mathrm{wBC}_s(H, \vec{x}_H)\\ &= \left\{ \vec{y}_H \in \{0,1\}^H \;\middle|\; \begin{array}{l} \exists v \;\big(\; (\forall j \in H : \vec{y}_{H|\{j\}} = v) \\ \wedge \; (H = \mathcal{P} \Rightarrow v = \vec{x}_{H|\{s\}}) \; \big) \end{array} \right\}.\end{aligned}$$

It was shown in [11] that weak broadcast cannot be constructed from authenticated channels using a deterministic protocol.

▶ **Lemma 24.** *[11] There exists no deterministic r-round protocol $\mathbf{\Pi}$ which allows for* $\{\textsc{Auth}\} \longrightarrow \mathrm{wBC}_i$.

Proof. Without loss of generality, let P_1 be the sender. Suppose there exists a deterministic r-round protocol $\mathbf{\Pi}$ which allows to construct wBC_1 from AUTH. Then, there exist protocol systems $\mathbf{\Pi}_1^x, \mathbf{\Pi}_2, \mathbf{\Pi}_3$ for parties P_1, P_2, P_3, where x denotes the input of P_1. Choose $k > r + 1$ as a multiple of 3 and arrange $4k$ such systems in a ring as follows: Start with a system $\mathbf{\Pi}_1^0$ and continue with systems $\mathbf{\Pi}_2, \mathbf{\Pi}_3$; each system is connected via authenticated channels to its predecessor and successor. Now repeat this pattern going clockwise, until $2k$ systems have been connected in this manner. Because k is a multiple of three, the last system in this arrangement will be a system $\mathbf{\Pi}_3$. Now, restart the pattern from the end of this arrangement, but instead of $\mathbf{\Pi}_1^0$, use $\mathbf{\Pi}_1^1$. Arrange another $2k$ nodes in this manner, thereby closing the ring.

Consider the system $\mathbf{\Pi}_1^0$ at "the top" of the ring. As all systems in the ring are deterministic the view of $\mathbf{\Pi}_1^0$ after r rounds is the same as if the system were run in a triangular configuration (where the triangle consists of $\mathbf{\Pi}_1^0, \mathbf{\Pi}_2, \mathbf{\Pi}_3$). The validity of weak broadcast implies that the system $\mathbf{\Pi}_1^0$ must output 0. Similarly, the system $\mathbf{\Pi}_1^1$ at "the bottom" of the ring must output 1. Now, consider any to adjacent systems in the ring. One can view the rest of the ring as an attack strategy of a corrupted party. Thus by consistency of weak broadcast any two adjacent systems must output the same value. We thus arrive at a contradiction. ◀

On the other hand, the results of [3] imply that weak broadcast can be achieved from authenticated channels for any $\epsilon > 0$.

▶ **Lemma 25.** *[3] For any $\epsilon > 0$ $\{\text{AUTH}\} \xrightarrow{\epsilon} \text{wBC}_i$.*

Finally, we show that weak broadcast is separated from broadcast. More precisely, we show that broadcast allows to construct weak broadcast while on the other hand broadcast cannot be constructed from weak broadcast within $\epsilon \leq \frac{1}{3}$.

▶ **Theorem 26.** *Weak broadcast and broadcast are strongly separated.*

The theorem follows from the following two lemmata.

▶ **Lemma 27.** *For all $i \in \{1,2,3\}$ $\{\text{BC}_i\} \longrightarrow \text{wBC}_i$.*

Proof. For all H and all $\vec{x}_H$ it holds that $\text{BC}_i(H, \vec{x}_H) \subseteq \text{wBC}_i(H, \vec{x}_H)$. This directly implies $\{\text{BC}_i\} \longrightarrow \text{wBC}_i$. ◀

▶ **Lemma 28.** *For all $i \in \{1,2,3\}$ $\{\text{wBC}_i, \text{AUTH}\} \overset{\frac{1}{3}}{\not\longrightarrow} \text{BC}_i$.*

Proof. We first show that $\text{XC}_{i,j}$ for $j \neq i$ is enough to construct wBC_i. The following protocol π allows P_i to weak broadcast its bit b using $\text{XC}_{i,j}$.

1. $\text{XC}_{i,j}$ is invoked where P_i inputs b and P_j inputs 0. Denote by b_i, b_j, b_k the bits the parties P_i, P_j, P_k receive as output from $\text{XC}_{i,j}$.
2. P_i outputs b_i, P_j outputs b_j and P_k outputs b_k.

The properties of $\text{XC}_{i,j}$ ensure that honest parties will always output the same bit, as required by the consistency of wBC_i. If at least P_i and P_j are honest, the output of $\text{XC}_{i,j}$ is $b = b \oplus 0$. The protocol thus achieves the validity condition required by wBC_i.

From Lemma 21, we know that $\{\text{XC}_{i,j}, \text{AUTH}\} \overset{\frac{1}{3}}{\not\longrightarrow} \text{BC}_i$. This implies that BC_i cannot be constructed from $\{\text{wBC}_i, \text{AUTH}\}$ within $\epsilon \leq \frac{1}{3}$. ◀

In summary, considering constructions for $\epsilon > 0$, weak broadcast is not stronger than authenticated channels. It is only when considering perfect constructions that weak broadcast provides strictly stronger guarantees. This is in contrast to XOR-cast which is stronger than authenticated channels for any $\epsilon \geq 0$.

5 Extension to the n-Party Case

In this section, we show how our theorems can be generalized to the n-party case. Note that our formal definition of XOR-cast can be used without modification for the setting with n parties. An informal explanation of the resulting specification is as follows. Again, parties P_i and P_j each input bits b_i and b_j. As in the three-party setting, if all parties behave correctly, the value of $b_i \oplus b_j$ should be output by all parties. If one or both of the parties P_i, P_j is dishonest, the honest parties should output the same value. In any other case, the remaining honest parties should output $b_i \oplus b_j$.

We begin by proving an n-party analogon of Theorem 15. Informally, we prove that, given any set of at most $\frac{2n}{3}$ distinct broadcast channels, no further broadcast channels can be achieved.

▶ **Theorem 29.** *Let* $\mathcal{B} = \{\mathrm{BC}_{\frac{n}{3}+1}, ..., \mathrm{BC}_n\}$. *Then* $\{\textsc{Auth}\} \cup \mathcal{B} \overset{\frac{1}{3}}{\not\rightarrow} \mathrm{BC}_1$.

Proof. We show that the existence of such a protocol would contradict Corollary 16. Thus, assume that there exists a protocol $\mathbf{\Pi}$ which allows to construct BC_k from $\{\textsc{Auth}\} \cup \mathcal{B}$ within $\epsilon = \frac{1}{3}$. In particular, $\forall H' \subseteq \mathcal{P}$ of size $\frac{2n}{3}$ we have that

$$\mathsf{P}\big(\mathbf{\Pi}(\textsc{Auth}, \mathcal{B})(H, \vec{x}_{H'}) \subseteq \mathrm{BC}_1(H', \vec{x}_{H'})\big) > 1 - \epsilon. \qquad (2)$$

We show now that this implies the existence of a protocol $\mathbf{\Pi}'$ which allows to construct BC_1 within $\frac{1}{3}$ in the three-party setting. In particular, for protocol $\mathbf{\Pi}'$ it will hold that $\forall H \subseteq \{P_1, P_2, P_3\}$ of size two that $\mathsf{P}\big(\mathbf{\Pi}'(\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3)(H, \vec{x}_H) \subseteq \mathrm{BC}_1(H, \vec{x}_H)\big) > 1 - \epsilon$, which is a direct contradiction of Corollary 16.

The idea of $\mathbf{\Pi}'$ is to execute protocol $\mathbf{\Pi}$ where each of the three parties P_1, P_2, P_3 emulates $\frac{n}{3}$ of the n parties. Concretely, party P_1 emulates virtual parties $P_1, ..., P_{\frac{n}{3}}$, party P_2 emulates $P_{\frac{n}{3}+1}, ..., P_{\frac{2n}{3}}$, and party P_3 emulates $P_{\frac{2n}{3}+1}, ..., P_n$. Clearly, all communication between virtual parties that occurs over authenticated channels can easily be emulated. Similarly, if a party $P_i, i \in \{\frac{n}{3}+1, ..., n\}$ broadcasts in $\mathbf{\Pi}$, then the party P_2 or P_3 emulating P_i can use BC_2 or BC_3, respectively, to carry out P_i's virtual broadcast over BC_i.

We can now map the set of real honest parties to sets of virtual honest parties. For instance, for $H = \{P_1, P_2\}$, the virtual parties in $H_1' = \left\{P_1, \ldots, P_{\frac{2n}{3}}\right\}$ are honest. Similarly, for $H = \{P_1, P_3\}$ and $H = \{P_21, P_3\}$ we have virtual honest sets H_2' and H_3', respectively. By the initial assumptions, in particular the one in Equation 2, it thus follows that $\mathsf{P}\big(\mathbf{\Pi}'(\textsc{Auth}, \mathrm{BC}_2, \mathrm{BC}_3)(H, \vec{x}_H) \subseteq \mathrm{BC}_1(H, \vec{x}_H)\big) > 1 - \epsilon$ for any H of size two. But this contradicts Corollary 16. ◀

In a similar fashion, one can prove the following statement for the n-party case.

▶ **Lemma 30.** $\{\textsc{Auth}\} \overset{\frac{1}{4}}{\not\rightarrow} \mathrm{XC}_{1,2}$.

Also, using almost the same arguments, we can prove the analogue of Lemma 20.

▶ **Lemma 31.** *For all* $i \neq j \in [n]$ $\{\textsc{Auth}, \mathrm{BC}_i\} \longrightarrow \mathrm{XC}_{i,j}$.

Finally, we can also restate Lemma 21 for the n-party case. Like the previous two lemmata, the proof proceeds in a similar fashion as the proof for the three-party case.

▶ **Lemma 32.** $\{\mathrm{XC}_{1,2}, \mathrm{XC}_{1,3}, \mathrm{XC}_{2,3}, \textsc{Auth}\} \overset{\frac{1}{3}}{\not\rightarrow} \mathrm{BC}_1$.

6 Conclusion and Outlook

In this work, we showed strong separation results between broadcast and authenticated channels. In particular, we showed that weak broadcast admits a strong separation from broadcast. In order to derive these separations, we generalized known techniques for proving impossibility to cover also probabilistic constructions. We believe that the formal techniques and the framework that we introduced here will prove useful to future efforts in proving similar results. We also initiated the natural study of *asymmetric consistency primitives*, in which a (strict) subset of the parties has input and every party receives output. Although both broadcast and weak broadcast are examples of such primitives, our work is the first to consider primitives in which the subset of parties with input is not a singleton set. We show that for the example of the XOR-cast, this type of consistency primitive falls into a previously undiscovered intermediate layer between authenticated channels and broadcast. As such, we believe that our work opens up several interesting lines of future research. In regards to further extending the scope of impossibility results, it would be interesting to see whether our techniques for probabilistic constructions can also be used to derive stronger bounds in settings with more complicated setup such as [4, 1]. Another interesting direction for future research would be a closer study of asymmetric consistency primitives in the above sense. A first question in this area would be to see if the hierarchy of three-party specifications considered in this work has an even deeper structure than outlined here, or, more generally, to classify all such specifications. A second immediate question would be to investigate how the picture changes when we consider primitives with more than three parties or when switching to stronger models of corruption, such as the *general adversary model* [7, 8, 16] (as opposed to the threshold setting we considered here). Conceptually, it would also be worthwhile to derive connections between such results and the field of information theoretic MPC.

References

1 Jeffrey Considine, Matthias Fitzi, Matthew K. Franklin, Leonid A. Levin, Ueli M. Maurer, and David Metcalf. Byzantine agreement given partial broadcast. *Journal of Cryptology*, 18(3):191–217, jul 2005.

2 Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. In Michael A. Malcolm and H. Raymond Strong, editors, *4th ACM Symposium Annual on Principles of Distributed Computing*, pages 59–70, Minaki, Ontario, Canada, aug 5–7, 1985. Association for Computing Machinery.

3 Matthias Fitzi, Daniel Gottesman, Martin Hirt, Thomas Holenstein, and Adam Smith. Detectable byzantine agreement secure against faulty majorities. In Aleta Ricciardi, editor, *21st ACM Symposium Annual on Principles of Distributed Computing*, pages 118–126, Monterey, California, USA, jul 21–24, 2002. Association for Computing Machinery.

4 Matthias Fitzi and Ueli M. Maurer. From partial consistency to global broadcast. In *32nd Annual ACM Symposium on Theory of Computing*, pages 494–503, Portland, Oregon, USA, may 21–23, 2000. ACM Press.

5 Ronald L. Graham and Andrew Chi-Chih Yao. On the improbability of reaching byzantine agreements (preliminary version). In *21st Annual ACM Symposium on Theory of Computing*, pages 467–478, Seattle, Washington, USA, may 15–17, 1989. ACM Press.

6 Martin Hirt, Ueli Maurer, and Pavel Raykov. Broadcast amplification. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 419–439, San Diego, CA, USA, feb 24–26, 2014. Springer, Berlin, Germany. doi:10.1007/978-3-642-54242-8_18.

7 Martin Hirt and Ueli M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, 2000. Extended abstract in *Proc. 16th of ACM PODC '97.*

8 Martin Hirt and Daniel Tschudi. Efficient general-adversary multi-party computation. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 181–200, Bengalore, India, dec 1–5, 2013. Springer, Berlin, Germany. doi:10.1007/978-3-642-42045-0_10.

9 Alexander Jaffe, Thomas Moscibroda, and Siddhartha Sen. On the price of equivocation in byzantine agreement. In Darek Kowalski and Alessandro Panconesi, editors, *31st ACM Symposium Annual on Principles of Distributed Computing*, pages 309–318, Funchal, Madeira, Portugal, jul 16–18, 2012. Association for Computing Machinery.

10 Anna Rochelle Karlin and Andrew Chi-Chih Yao. Probabilistic lower bounds for the byzantine generals problem. unpublished manuscript, 1984.

11 Leslie Lamport. The weak byzantine generals problem. *Journal of the ACM*, 30(3):668–676, jul 1983.

12 Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, jul 1982.

13 Julian Loss, Ueli Maurer, and Daniel Tschudi. Hierarchy of three-party consistency specifications. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 3048–3052. IEEE, 2016.

14 Ueli Maurer. Towards a theory of consistency primitives. In Rachid Guerraoui, editor, *International Symposium on Distributed Computing — DISC 2004*, volume 3274 of *Lecture Notes in Computer Science*, pages 379–389. Springer, Berlin, Germany, 2004.

15 D. V. S. Ravikant, Venkitasubramaniam Muthuramakrishnan, V. Srikanth, K. Srinathan, and C. Pandu Rangan. On byzantine agreement over (2,3)-uniform hypergraphs. In Rachid Guerraoui, editor, *International Symposium on Distributed Computing — DISC 2004*, volume 3274 of *Lecture Notes in Computer Science*, pages 450–464. Springer, Berlin, Germany, Oct 2004.

16 Pavel Raykov. Broadcast from minicast secure against general adversaries. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP 2015: 42nd International Colloquium on Automata, Languages and Programming, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 701–712, Kyoto, Japan, jul 6–10, 2015. Springer, Berlin, Germany. doi:10.1007/978-3-662-47666-6_56.