# Hierarchy of Three-Party Consistency Specifications

Julian Loss, Ueli Maurer, Daniel Tschudi
Department of Computer Science, ETH Zurich, Switzerland
jloss@student.ethz.ch,{maurer, tschudid}@inf.ethz.ch

***Abstract*—In the theory of distributed systems and in cryptography one considers a set of $n$ parties which wish to 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. A well-known result [LSP82] states that this is possible if and only if strictly less than a third of the parties are dishonest.**

**Broadcast guarantees a very strong form of consistency. This paper investigates generalizations of the broadcast setting in two directions: weaker forms of consistency guarantees are considered, and other resources than merely bilateral channels are assumed to be available. The ultimate goal of this line of work is to arrive at a complete classification of consistency specifications [Mau04]. As a concrete result in this direction we present a complete classification of three-party specifications with a binary input and binary outputs.**

## I. Introduction

A Secure Multi-Party Computation (MPC) protocol allows a set of parties to securely perform an arbitrary computation on their inputs even when a subset of the parties is dishonest. Such a protocol can thus be seen as the emulation of a trusted party which performs the desired computation. One of the most fundamental primitives in this setting is broadcast. It allows a sender to distribute his message $m$ such that all honest parties receive the same value $m'$ (consistency) where $m' = m$ for an honest sender (validity). It is well-known [LSP82] that broadcast can only be constructed from authenticated channels if less than a third of all parties are dishonest. This raises questions such as "What is required to construct broadcast if more parties are dishonest?" and "What are the consistency guarantees one can achieve with authenticated channels?". A thorough answer can be given through a classification of so called consistency specifications.

### A. Summary of Known Results

Broadcast was first introduced by Pease, Shostak, and Lamport [LSP82]. They showed that one cannot construct broadcast from a network of pairwise authenticated channels if a third or more of the parties are dishonest. The feasibility of broadcast given authenticated channels has been intensively studied (see e.g. [PW96],[BGP89]) and protocols with optimal resilience and complexity have been proposed. The work in [FM00] considers the problem of constructing global broadcast given broadcast among any three parties.

An essential characteristic of broadcast are the consistency guarantees that it provides on the output of honest parties. To model general consistency guarantees, Maurer [Mau04] proposed the notion of consistency specifications. A consistency specification defines for every set $H$ of honest parties and every possible input of those parties, a consistency guarantee on their output. Protocols allow to construct (strong) consistency specifications from a given set of (weak) consistency specifications. The notion of consistency specifications does not allow to model secrecy requirements. A consistency specification thus corresponds to a trusted party which enables dishonest parties to learn the inputs of honest parties. For a more general setting where secrecy matters, one may use security frameworks such as [Can01] or [MR11] and [Mau11].

### B. Contributions and Outline

In this work, we investigate the separation between broadcast and authenticated channels by considering a classification of consistency specifications. Given a set of consistency specifications, one can consider its closure, i.e., all consistency specifications which one can construct from this set. This leads to a natural classification where two consistency specifications are in the same class if they have the same closure. We proceed as follows. First, we revisit the notion of consistency specifications from [Mau04] and provide rigorous definitions of the basic concepts. Then we introduce different flavors of constructions and define a means of classification for consistency specifications. In a second part we give a complete classification of three-party specifications where a fixed party can give a binary input. This provides some surprising insight into the structure and hierarchy of these specifications.

## II. 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 tuple of sets $(M_1, \ldots, M_n)$ and a subset $S \subseteq \mathcal{P}$, we denote by $M_S$ the Cartesian product $\bigtimes_{i \in S} M_i$. We denote by $v_S|_{S'} \in M_{S'}$ the projection of $v_S \in M_S$ to entries in $S' \subseteq S$. For a subset $L \subseteq M_S$ we can similarly define $L|_{S'} := \{v_S|_{S'} \mid v_S \in L\}$. Moreover we write $[n]$ for the set $\{1, \ldots, n\}$.

### A. Consistency specification

In the following we consider primitives and protocols where each party $P_i$ has an input from a finite input domain $\mathcal{D}_i$ and receives an output from a finite output domain $\mathcal{R}_i$. We assume that both $\mathcal{D}_i$ and $\mathcal{R}_i$ are non-empty sets. If a party has no input and/or no output, the corresponding domains are assumed to be singletons containing the symbol $\perp$. To model consistency guarantees we use the notion of consistency specifications.

**Definition 1** ([Mau04])**.** *Consider non-empty $\mathcal{P}$, $\mathcal{D}_\mathcal{P}$, and $\mathcal{R}_\mathcal{P}$ as above. A $(\mathcal{P}, \mathcal{D}_\mathcal{P}, \mathcal{R}_\mathcal{P})$-*consistency specification *is a function assigning to every non-empty subset $H \subseteq \mathcal{P}$ and every input vector $x_H \in \mathcal{D}_H$ a non-empty set $\mathcal{C}(H, x_H) \subseteq \mathcal{R}_H$, satisfying the* monotonicity constraint*: For any non-empty subsets $H' \subseteq H \subseteq \mathcal{P}$*

$$\mathcal{C}(H, x_H)|_{H'} \subseteq \mathcal{C}(H', x_{H|H'}). \tag{1}$$

Our definition of a consistency specification differs slightly from the original version in [Mau04]. First, if all parties are dishonest, i.e., $H = \varnothing$, there are no consistency guarantees to be formulated. We therefore restrict the domain of consistency specifications to non-empty subsets $H \subseteq \mathcal{P}$. Secondly, we require that the honest parties are guaranteed at least one output, i.e., $\varnothing \neq \mathcal{C}(H, x_H)$ for any $H$ and $x_H$.

**Example 1.** *An* authenticated channel *from $P_i$ to $P_j$ is a consistency specification, denoted as* $\text{AUTH}_{i,j}$*, where only $P_i$ has input (i.e., $\mathcal{D}_k = \{\perp\}$ for $k \neq i$) and where $P_j$'s output is equal to the input of $P_i$ if both of them are honest. There are no other consistency constraints on the outputs of honest parties. More formally, we have for all $H \subseteq \mathcal{P}$ and $x_H \in \mathcal{D}_H$:*

$$\text{AUTH}_{i,j}(H, x_H) = \big\{ y_H \in \mathcal{R}_H | i, j \in H \Rightarrow y_{H|\{j\}} = x_{H|\{i\}} \big\}$$

**Example 2.** *A* broadcast channel *for party $P_i$ is denoted as* $\text{BC}_i$*. The formal definition is as follows.*

$$\begin{aligned}\text{BC}_i(H, x_H) = \big\{ y_H \in \mathcal{R}_H \mid \exists v \big( & (\forall j \in H \setminus \{i\} : y_{H|\{j\}} = v) \\ & \wedge (i \in H \Rightarrow v = x_{H|\{i\}}) \big) \big\}\end{aligned}$$

*for all $H \subseteq \mathcal{P}$ where $\mathcal{D}_k = \{\perp\}$ for $k \neq i$.*

In order to compare consistency specifications and the strength of their consistency guarantees, one can use the following natural (partial) ordering.

**Definition 2.** *Consider two $(\mathcal{P}, \mathcal{D}_\mathcal{P}, \mathcal{R}_\mathcal{P})$-consistency specifications $\mathcal{C}_1$ and $\mathcal{C}_2$. Then $\mathcal{C}_2$ is* stronger *than $\mathcal{C}_1$, denoted $\mathcal{C}_1 \preceq \mathcal{C}_2$, if $\mathcal{C}_2(H, x_H) \subseteq \mathcal{C}_1(H, x_H)$ for all non-empty subsets $H \in \mathcal{P}$ and every $x_H \in \mathcal{D}_H$.*

## III. Protocols and Constructions

### A. Basic Constructions

A fundamental aspect of computer science is to realize complex objects from simpler ones. In our context this leads to the question whether one can construct a certain consistency specification from weaker consistency specifications by means of a protocol. A protocol execution consists of several rounds where in each round $j$ a consistency specification $\mathcal{C}^{(j)}$ is invoked. Each party $P_i$ computes its input to $\mathcal{C}^{(j)}$ as a function $f_i^{(j)}$ of its protocol input and the outputs it received from previously invoked consistency specifications. At the end each party $P_i$ computes its protocol output as a function $g_i$ of its input and all the outputs it received from invoked specifications. More formally, a protocol is defined as a tuple of functions. For $\ell \geq 0$ let $\vec{\mathcal{C}} = (\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(\ell)})$ be an $\ell$-tuple of consistency specifications where $\mathcal{C}^{(j)}$ is a $(\mathcal{P}, \mathcal{D}_\mathcal{P}^{(j)}, \mathcal{R}_\mathcal{P}^{(j)})$-consistency specification for $j \in [\ell]$.

**Definition 3** ([Mau04])**.** *An* $\ell$-round protocol $\pi$ suitable *for $\vec{\mathcal{C}}$ with input domains $\mathcal{D}_\mathcal{P}$ and output domains $\mathcal{R}_\mathcal{P}$ consists of a tuple of functions $(f_i^{(j)}, g_i)$ for $i \in \mathcal{P}$ and $j \in [\ell]$ where*

$$f_i^{(j)} : \mathcal{D}_i \times \mathcal{R}_i^{(1)} \times \cdots \times \mathcal{R}_i^{(j-1)} \to \mathcal{D}_i^{(j)}$$

*and*

$$g_i : \mathcal{D}_i \times \mathcal{R}_i^{(1)} \times \cdots \times \mathcal{R}_i^{(\ell)} \to \mathcal{R}_i.$$

For convenience we will use $f_H^{(j)}$ or $g_H$ to denote the parallel composition[1] of the corresponding functions $f_i, g_i$ for $i \in H$. We remark that a protocol only refers to input and output domains. It does not define the particular specifications which are invoked during an execution of the protocol. Note that we allow zero-round protocols, i.e., protocols where no specification is invoked. Given a tuple of $\vec{\mathcal{C}}$ of consistency specifications and a suitable protocol $\pi$, the execution of $\pi$ on $\vec{\mathcal{C}}$ constructs a new consistency specification.

**Definition 4.** *A suitable protocol $\pi$ for tuple $\vec{\mathcal{C}}$* constructs *specification $\mathcal{C}$ from $\vec{\mathcal{C}}$, denoted $\vec{\mathcal{C}} \xrightarrow{\pi} \mathcal{C}$, if $\mathcal{C} \preceq \pi\vec{\mathcal{C}}$ where*

$$\begin{aligned}\pi\vec{\mathcal{C}}(H, x_H) = \big\{ & y_H \in \mathcal{R}_H \bigm| \forall j \in [\ell]\ \exists y_H^{(j)} \in \mathcal{R}_H^{(j)} : \\ & y_H^{(j)} \in \mathcal{C}^{(j)}\big(H, f_H^{(j)}(x_H, y_H^{(1)}, \ldots, y_H^{(j-1)})\big) \\ & \wedge\ y_H = g_H(x_H, y_H^{(1)}, \ldots, y_H^{(\ell)}) \big\}\end{aligned}$$

*for all non-empty subsets $H \in \mathcal{P}$ and every $x_H \in \mathcal{D}_H$.*

With this basic type of construction we can now introduce derived types of constructions.

### B. Constructions from Sets

In the setting of distributed computing it is common to assume that parties are not restricted in the use of given primitives. It is thus natural to consider constructions from a set $\mathfrak{C}$ of consistency specifications where each specification in $\mathfrak{C}$ may be invoked arbitrarily often during a protocol execution.

**Definition 5.** *Let $\mathfrak{C}$ be a set of consistency specifications. A protocol $\pi$* constructs *specification $\mathcal{C}$ from $\mathfrak{C}$, denoted as*

$$\mathfrak{C} \xrightarrow{\pi} \mathcal{C},$$

*if there exist a tuple $\vec{\mathcal{C}}$ over $\mathfrak{C}$ such that $\vec{\mathcal{C}} \xrightarrow{\pi} \mathcal{C}$.*

If a construction from $\mathfrak{C}$ to $\mathcal{C}$ exists, we simply write $\mathfrak{C} \longrightarrow \mathcal{C}$, and $\mathfrak{C} \not\longrightarrow \mathcal{C}$ otherwise. A set of consistency specifications $\mathfrak{C}'$ is constructible from $\mathfrak{C}$, denoted by $\mathfrak{C} \longrightarrow \mathfrak{C}'$ if all $\mathcal{C} \in$

[1]Formally, for any $i \in H$, $x_H \in \mathcal{D}_H$ and $y_H^{(j)} \in \mathcal{R}_H^{(j)}$ we have that

$$f_H^{(j)}(x_H, y_H^{(1)}, \ldots, y_H^{(j-1)})|_i = f_i^{(j)}(x_H|_i, y_H^{(1)}|_i, \ldots, y_H^{(j-1)}|_i).$$

$\mathfrak{C}'$ can be constructed from $\mathfrak{C}$. We note that this notion of construction is transitive in the following sense.

**Lemma 1.** *Let $\mathfrak{C}_1, \mathfrak{C}_2, \mathfrak{C}_3$ be sets of consistency specifications for $\mathcal{P}$. Suppose $\mathfrak{C}_1 \longrightarrow \mathfrak{C}_2$ and $\mathfrak{C}_2 \longrightarrow \mathfrak{C}_3$. Then $\mathfrak{C}_1 \longrightarrow \mathfrak{C}_3$.*

The closure of a consistency specification set $\mathfrak{C}$ with respect to a consistency specification set $\mathfrak{T}$ contains all specifications in $\mathfrak{T}$ which can be constructed from $\mathfrak{C}$.

**Definition 6.** *The (relative)* closure *of $\mathfrak{C}$ with respect to $\mathfrak{T}$ is defined as $\langle \mathfrak{C} \rangle_{\mathfrak{T}} := \{\mathcal{C} \in \mathfrak{T} \mid \mathfrak{C} \longrightarrow \mathcal{C}\}$.*

The closure is monotone, i.e., for $\mathfrak{C}' \subseteq \mathfrak{C}$ it holds that $\langle \mathfrak{C}' \rangle_{\mathfrak{T}} \subseteq \langle \mathfrak{C} \rangle_{\mathfrak{T}}$. We will omit $\mathfrak{T}$, if clear from the context, and simply write $\langle \mathfrak{C} \rangle$. To classify a collection $\mathbf{S} = \{\mathfrak{C}_1, \ldots, \mathfrak{C}_k\}$ of consistency specification sets with respect to $\mathfrak{T}$ one can consider the different closures of the sets in $\mathbf{S}$.

**Definition 7.** *A* classification *of a collection $\mathbf{S}$ of consistency specification sets with respect to $\mathfrak{T}$ is the set $\{\langle \mathfrak{C} \rangle_{\mathfrak{T}} \mid \mathfrak{C} \in \mathbf{S}\}$. Two sets $\mathfrak{C}, \mathfrak{C}' \in \mathbf{S}$ realize the same class if $\langle \mathfrak{C} \rangle_{\mathfrak{T}} = \langle \mathfrak{C}' \rangle_{\mathfrak{T}}$.*

### C. Constructions from Multisets

The assumption that parties can invoke given consistency specifications arbitrary often is rather strong. One can therefore consider a weaker type of construction where a given specification can be used a limited number of times. Formally, this leads to constructions from multisets where the multiplicity of an element bounds the number of its invocation during a protocol execution. A tuple $\vec{\mathcal{C}}$ over multi-set $\mathscr{C}$ is therefore suitable if the occurrence of an element $\mathcal{C} \in \mathscr{C}$ in $\vec{\mathcal{C}}$ is bounded by its multiplicity.

**Definition 8.** *Let $\mathscr{C}$ be a multiset of consistency specifications. A protocol $\pi$* constructs *specification $\mathcal{C}$ from $\mathscr{C}$, denoted as*

$$\mathscr{C} \xrightarrow{\pi} \mathcal{C},$$

*if there exist a tuple $\vec{\mathcal{C}}$ suitable both for $\pi$ and $\mathscr{C}$ such that $\vec{\mathcal{C}} \xrightarrow{\pi} \mathcal{C}$.*

The possibility of a multiset construction implies the possibility of a normal (set) construction, i.e., $\mathscr{C} \xrightarrow{\pi} \mathcal{C}$ implies $\mathfrak{C} \xrightarrow{\pi} \mathcal{C}$ for the underlying set $\mathfrak{C}$ of the elements in $\mathscr{C}$.

## IV. Classification of 3-Party Specifications

The goal of this section is to provide a motivating example of a consistency specifications classification. For this purpose we consider specifications for three parties $\mathcal{P} = \{P_1, P_2, P_3\}$ where $P_1$ has a binary input and the other parties have binary outputs.

**Definition 9.** *The set of* binary $P_1$-input 3-party *consistency specifications, denoted $\Omega_1$, consists of all $(\mathcal{P}, \{0,1\} \times \{\bot\}^2, \{\bot\} \times \{0,1\}^2)$-consistency specifications.*

We remark that the output of $P_1$ is empty which is equivalent to giving no consistency guarantees for $P_1$. However, this is not a real limitation as one can trivially construct any $(\mathcal{P}, \{0,1\} \times \{\bot\}^2, \{0,1\}^3)$-consistency specification using one from $\Omega_1$. In the following we will omit empty inputs and outputs, e.g. we will write $b \in \mathcal{C}(\{P_1, P_2\}, b)$ instead of $(\bot, b) \in \mathcal{C}(\{P_1, P_2\}, (\bot, b))$. For our classification we assume that parties are pairwise connected by authenticated channels and have access to a subset of specifications from $\Omega_1$. Formally we thus consider a classification of the collection $\mathbf{S} := \{\mathfrak{C} \cup \textsc{Auth} \mid \mathfrak{C} \subseteq \Omega_1\}$ with respect to $\mathfrak{T} := \Omega_1$ where $\textsc{Auth}$ is the set of all authenticated channels for $\mathcal{P}$.

### A. Complete Classification of $\Omega_1$

In this section we show that $\Omega_1$ is divided into two classes. First, we have the class $\langle \textsc{Auth} \rangle$ which consists of of all specifications which can be constructed from authenticated channels. Second, we have the complement $\Omega_1 \setminus \langle \textsc{Auth} \rangle$ which consists of all specifications which allow to construct broadcast given authenticated channels. In a first step, we derive a sufficient and necessary condition for $\mathcal{C} \in \langle \textsc{Auth} \rangle$ by considering the following sets of binary tuples. Let $M_{\mathcal{C}} = \mathcal{C}(\{P_2, P_3\})$ and for any $x \in \{0,1\}$ let $M_{2,\mathcal{C}}^{(x)} = \{(y_2, y_3) \mid y_2 \in \mathcal{C}(\{P_1, P_2\}, x)\}$, and $M_{3,\mathcal{C}}^{(x)} = \{(y_2, y_3) \mid y_3 \in \mathcal{C}(\{P_1, P_3\}, x)\}$.

**Lemma 2.** *A specification $\mathcal{C} \in \Omega_1$ can be constructed from authenticated channels, i.e., $\mathcal{C} \in \langle \textsc{Auth} \rangle$, if $M_{2,\mathcal{C}}^{(0)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(1)} \neq \varnothing$ and $M_{2,\mathcal{C}}^{(1)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(0)} \neq \varnothing$.*

*Proof.* To construct $\mathcal{C}$ from authenticated channels consider the following protocol $\pi$. First, party $P_1$ sends its input bit

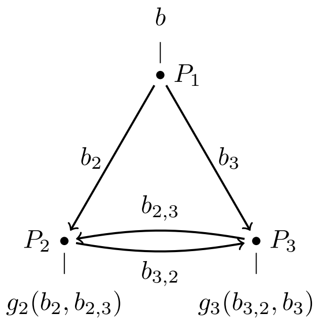



Figure 1: Protocol for Lemma 2.

to the other parties which exchange the received bits (cf. Figure 1). The output of $P_2$ is $g_2(b_2, b_{2,3})$ for a function $g_2 : \{0,1\}^2 \to \{0,1\}$ where $b_2$ and $b_{2,3}$ are the bits received from $P_1$ and $P_3$. Analogously, $P_3$ outputs $g_3(b_{3,2}, b_3)$. The assumption of the Lemma allows us to define $g_2$ and $g_3$ as follows. For any bit $b \in \{0,1\}$ let

$$\big(g_2(b,b), g_3(b,b)\big) \in \mathcal{C}(\{P_1, P_2, P_3\}, b) \tag{2}$$

and

$$\big(g_2(b,1-b), g_3(b,1-b)\big) \in M_{2,\mathcal{C}}^{(b)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(1-b)}. \tag{3}$$

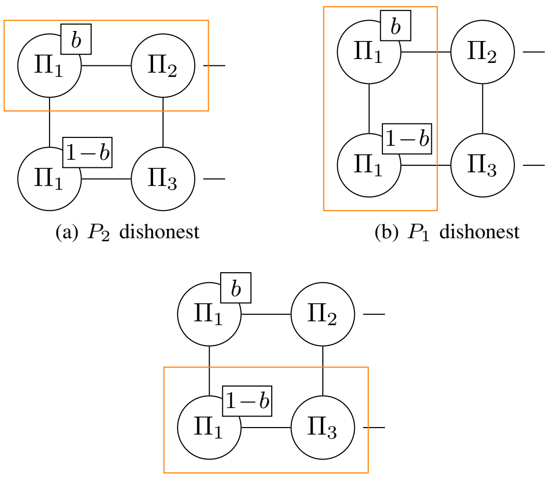
Note that for any $b \in \{0,1\}$ we have $\mathcal{C}(\{P_1, P_2, P_3\}, b) \subseteq M_{2,\mathcal{C}}^{(b)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(b)}$ and thus $M_{2,\mathcal{C}}^{(x)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(y)} \neq \varnothing$ for any $x, y \in \{0,1\}$. Consider now the following cases. If everyone is honest, we have $b = b_2 = b_3 = b_{2,3} = b_{3,2}$. The output of $P_2$ and $P_3$ is thus $(g_2(b,b), g_3(b,b)) \in \mathcal{C}(\{P_1, P_2, P_3\}, b)$.

For $H = \{P_1, P_2\}$ we have $b_2 = b$ and the output of $P_2$ is therefore $g_2(b, b_{2,3}) \in (M_{2,\mathcal{C}}^{(b)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(b_{2,3})})|_{P_2}$. Thus by the definition of $M_{2,\mathcal{C}}^{(b)}$ it holds that $g_2(b, b_{2,3}) \in \mathcal{C}(\{P_1, P_2\}, b)$. For $H = \{P_1, P_3\}$ it similarly holds that $b_3 = b$ and $g_3(b_{3,2}, b) \in \mathcal{C}(\{P_1, P_3\}, b)$. If $H = \{P_2, P_3\}$, it holds that $b_2 = b_{3,2}$ and $b_3 = b_{2,3}$. The output of the parties is therefore $(g_2(b_2, b_3), g_3(b_2, b_3)) \in M_{2,\mathcal{C}}^{(b_2)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(b_3)}$ and thus $(g_2(b_2, b_3), g_3(b_2, b_3)) \in M_{\mathcal{C}} = \mathcal{C}(\{P_2, P_3\})$. All the other cases follow directly from the monotonicity of $\mathcal{C}$. The protocol $\pi$ therefore constructs $\mathcal{C}$ from authenticated channels. $\square$

The next Lemma shows that the above condition is also necessary for $\mathcal{C} \in \langle \text{AUTH} \rangle$.

**Lemma 3.** *A specification $\mathcal{C} \in \Omega_1$ with $M_{2,\mathcal{C}}^{(0)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(1)} = \varnothing$ or $M_{2,\mathcal{C}}^{(1)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(0)} = \varnothing$ cannot be constructed from authenticated channels, i.e., $\mathcal{C} \notin \langle \text{AUTH} \rangle$.*

*Proof.* The following proof is generalization of a proof technique in [FLM85]. Consider a $\mathcal{C} \in \Omega_1$ with $M_{2,\mathcal{C}}^{(b)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(1-b)} = \varnothing$ for a $b \in \{0, 1\}$. Towards a contradiction, let us assume that there exists a protocol $\pi$ such that $\text{AUTH} \xrightarrow{\pi} \mathcal{C}$. Then there exist (deterministic) systems $\Pi_1, \Pi_2, \Pi_3$ for parties $P_1, P_2, P_3$. Each system executes the protocol part of the corresponding parties and can be connected to two other systems. Consider a dishonest $P_2$ emulating $\Pi_1$ and $\Pi_2$ in




(a) $P_2$ dishonest

(b) $P_1$ dishonest

(c) $P_3$ dishonest

Figure 2: Corruption Scenarios for Lemma 3.

a normal protocol execution as in Figure 2a where a bit in a box next to a system denotes the system's input. As the protocol constructs $\mathcal{C}$ system $\Pi_3$ must output a bit $b_3$ in $\mathcal{C}(\{P_1, P_3\}, 1 - b)$. Next assume that a dishonest $P_1$ emulates $\Pi_1$ twice but with different inputs as in Figure 2b. In this case the output tuple $(b_2, b_3)$ of systems $\Pi_2$ and $\Pi_3$ must be in $\mathcal{C}(\{P_2, P_3\})$. Lastly, suppose that a dishonest $P_3$ emulates $\Pi_1$ and $\Pi_3$ as in Figure 2c. Here the output bit $b_2$ of $\Pi_2$ must be in $\mathcal{C}(\{P_1, P_2\}, b)$. Note that those three cases describe the same combined system which outputs bit $b_2$ at $\Pi_2$ and bit $b_3$ at $\Pi_3$. It follows that $(b_2, b_3) \in M_{2,\mathcal{C}}^{(b)} \cap M_{\mathcal{C}} \cap M_{3,\mathcal{C}}^{(1-b)} = \varnothing$, a contradiction. Therefore there exists no protocol constructing $\mathcal{C}$ from authenticated channels. $\square$

We already know that broadcast $\text{BC}_1$ can not be constructed from authenticated channels. Moreover, observe that any specification in $\Omega_1$ can be constructed from broadcast.

**Lemma 4.** *For any $\mathcal{C} \in \Omega_1$, $\{\text{BC}_1\} \longrightarrow \mathcal{C}$.*

*Proof.* The construction is trivially achieved with the following protocol. First, $P_1$ broadcasts its input $b$. Then parties $P_2, P_3$ output some fixed tuple in $\mathcal{C}(\{P_1, P_2, P_3\}, b)$. $\square$

In the following we will show that any specification in $\mathcal{C} \in \Omega_1 \setminus \langle \text{AUTH} \rangle$ in addition to authenticated channels is enough to construct broadcast $\text{BC}_1$. The condition of Lemma 3 implies that any $\mathcal{C}' \in \Omega_1 \setminus \langle \text{AUTH} \rangle$ is equivalent[2] to a $\mathcal{C} \in \Omega_1 \setminus \langle \text{AUTH} \rangle$ with $\mathcal{C}(\{P_1, P_2, P_3\}, b) = \{(b, b)\}$ for $b \in \{0, 1\}$. This means that it is enough to henceforth $\mathcal{C}$ instead of $\mathcal{C}'$. Moreover, it follows from the Lemma that $2 \leq |\mathcal{C}(\{P_2, P_3\})| \leq 3$, $\mathcal{C}(\{P_1, P_2\}, 0) \neq \mathcal{C}(\{P_1, P_2\}, 1)$, and $\mathcal{C}(\{P_1, P_3\}, 0) \neq \mathcal{C}(\{P_1, P_3\}, 1)$. The above conditions imply that such a $\mathcal{C}$ is similar to broadcast $\text{BC}_1$ except that it offers (potentially) weaker consistency and validity guarantees (for $P_2$ or $P_3$). One can therefore describe the weakening by a triple in $\{\Diamond, 0, 1\}^3$ where $\Diamond$ means that the specific component is not weakened at all.

**Definition 10.** *Let $\alpha, \beta, \gamma \in \{\Diamond, 0, 1\}$. The* weak broadcast *$(\alpha, \beta, \gamma)$-$\text{wBC}_1$ is defined as follows:*

- $(\alpha, \beta, \gamma)\text{-wBC}_1(\{P_1, P_2, P_3\}, b) = \{(b, b)\}$
- $(\alpha, \beta, \gamma)\text{-wBC}_1(\{P_1, P_2\}, b) = \begin{cases} \{b\} & \text{if } b \neq \alpha \\ \{0,1\} & \text{if } b = \alpha \end{cases}$
- $(\alpha, \beta, \gamma)\text{-wBC}_1(\{P_1, P_3\}, b) = \begin{cases} \{b\} & \text{if } b \neq \beta \\ \{0,1\} & \text{if } b = \beta \end{cases}$
- $(\alpha, \beta, \Diamond)\text{-wBC}_1(\{P_2, P_3\}) = \{(0,0), (1,1)\}$
  $(\alpha, \beta, 0)\text{-wBC}_1(\{P_2, P_3\}) = \{(0,0), (0,1), (1,1)\}$
  $(\alpha, \beta, 1)\text{-wBC}_1(\{P_2, P_3\}) = \{(0,0), (1,0), (1,1)\}$

For example, weak broadcast $(\Diamond, \Diamond, 0)$-$\text{wBC}_1$ satisfies the validity condition of normal broadcast, but offers the weaker consistency guarantee $(\Diamond, \Diamond, 0)\text{-wBC}_1(\{P_2, P_3\}) = \{(0,0), (0,1), (1,1)\}$. Note also that $\text{BC}_1 = (\Diamond, \Diamond, \Diamond)\text{-wBC}_1$. The condition in Lemma 3 implies that some variants of weak broadcast can be constructed using authenticated channels.

**Lemma 5.** *For any $x \in \{\Diamond, 0, 1\}$ and $y \in \{0, 1\}$ the specifications $(x, y, y)$-$\text{wBC}_1$, $(y, x, 1-y)$-$\text{wBC}_1$, and $(y, y, x)$-$\text{wBC}_1$ are in $\langle \text{AUTH} \rangle$.*

*Proof.* Can be shown directly using Lemma 3. $\square$

It turns out that given authenticated channels one can construct $\text{BC}_1$ from rather weak variants of $\text{wBC}_1$. The first variant we consider is $(\Diamond, \Diamond, \gamma)$-$\text{wBC}_1$ for $\gamma \in \{\Diamond, 0, 1\}$ which is in $\Omega_1 \setminus \langle \text{AUTH} \rangle$ (cf. Lemma 3).

[2] Two specifications $\mathcal{C}'$ and $\mathcal{C}$ are equivalent if $\{\mathcal{C}'\} \longrightarrow \mathcal{C}$ and $\{\mathcal{C}\} \longrightarrow \mathcal{C}'$.

**Lemma 6.** *For any $\gamma \in \{\Diamond, 0, 1\}$,*

$$\textsc{Auth} \cup \{(\Diamond, \Diamond, \gamma)\text{-wBC}_1\} \longrightarrow \text{BC}_1.$$

*Proof.* For $\gamma = \Diamond$ we have $\text{BC}_1 = (\Diamond, \Diamond, \Diamond)\text{-wBC}_1$. For $\gamma \neq \Diamond$ and input bit $b$ of $P_1$ consider the following protocol. First, $P_1$ sends $(b, 1-b)$ to the other parties using two $(\Diamond, \Diamond, \gamma)$-wBC$_1$ invocations. Denote by $(b_2, c_2)$ (resp. $(b_3, c_3)$) the bits received by $P_2$ (resp. $P_3$). Then $P_2$ and $P_3$ exchange their bits using authenticated channels where $b_{2,3}, c_{2,3}$ (resp. $b_{3,2}, c_{3,2}$) denote the bits received by $P_2$ (resp. $P_3$). If $b_2 \neq c_2$, party $P_2$ outputs $b_2$. Otherwise, if $b_{2,3} \neq c_{2,3}$, $P_2$ outputs $b_{2,3}$. Otherwise $P_2$ outputs 0. The output of $P_3$ is computed analogous. Consider now the following cases. If everyone is honest, we have $b_2 = b_3 = b$ and $c_2 = c_3 = 1-b$. The output of $P_2, P_3$ is thus $(b_2, b_3) = (b, b)$. For $H = \{P_1, P_2\}$ we have $b_2 = b$ and $c_2 = 1-b$. The output of $P_2$ is therefore $b_2 = b$. For $H = \{P_1, P_3\}$ we have $b_3 = b$ and $c_3 = 1-b$. The output of $P_3$ is therefore $b_3 = b$. If $H = \{P_2, P_3\}$, we have $(b_{2,3}, c_{2,3}) = (b_3, c_3)$ and $(b_{3,2}, c_{3,2}) = (b_2, c_2)$. If $b_2 \neq b_3$, we have $c_2 = c_3$ as $(1-\gamma, \gamma) \notin (\Diamond, \Diamond, \gamma)\text{-wBC}_1$. It is now easy to check that $P_2$ and $P_3$ will output the same bit. □

Almost all weak-broadcast specifications where all three components are weakened are in $\langle \textsc{Auth} \rangle$ (cf. Lemma 5). The two exceptions are $(0,1,0)$-wBC$_1$ and $(1,0,1)$-wBC$_1$. Surprisingly one is able to construct broadcast from each of them given authenticated channels.

**Lemma 7.** $\textsc{Auth} \cup \{(0,1,0)\text{-wBC}_1\} \longrightarrow \text{BC}_1$ *and* $\textsc{Auth} \cup \{(1,0,1)\text{-wBC}_1\} \longrightarrow \text{BC}_1$.

*Proof.* With Lemma 6 it is enough to show that one can construct $(\Diamond, \Diamond, 0)$-wBC$_1$ from $(0,1,0)$-wBC$_1$ (resp. $(\Diamond, \Diamond, 1)$-wBC$_1$ from $(1,0,1)$-wBC$_1$. For $(\Diamond, \Diamond, 0)$-wBC$_1$ one can show this analogously to the proof of Lemma 6 using the following protocol.

First, $P_1$ sends its bit $b$ to the other parties using an authenticated channel, where $b_2$ (resp. $b_3$) denote the bit received by $P_2$ (resp. $P_3$). In the next step $P_1$ sends $b$ over $(0,1,0)$-wBC$_1$, where $c_2$ (resp. $c_3$) denotes the bit received by $P_2$ (resp. $P_3$). Finally, party $P_2$ outputs bit $o_2$ and party $P_3$ outputs bit $o_3$ where:

$$o_2 = \begin{cases} 1 & \text{if } (b_2, c_2) = (1,1) \\ 0 & \text{otherwise} \end{cases}$$

and

$$o_3 = \begin{cases} 0 & \text{if } (b_3, c_3) = (0,0) \\ 1 & \text{otherwise.} \end{cases}$$

□

Using the monotonicity of $\mathcal{C} \in \Omega_1 \setminus \langle \textsc{Auth} \rangle$ it is easy to show that one can either construct $(0,1,0)$-wBC$_1$ or $(1,0,1)$-wBC$_1$ from $\mathcal{C}$. With Lemma 7 this implies that given authenticated channels one can construct broadcast from $\mathcal{C}$.

**Lemma 8.** *Let $\mathcal{C} \in \Omega_1 \setminus \langle \textsc{Auth} \rangle$, then* $\textsc{Auth} \cup \{\mathcal{C}\} \rightarrow \text{BC}_1$.

We note that the availability of authenticated channels is crucial. For instance, one can show that $(\Diamond, \Diamond, 0)$-wBC$_1$ is strictly weaker than BC$_1$, i.e., $\{(\Diamond, \Diamond, 0)\text{-wBC}_1\} \not\rightarrow \text{BC}_1$. The following theorem summarizes our results.

**Theorem 1.** *Given authenticated channels and a set of specifications $\mathfrak{C} \subseteq \Omega_1$ one can either construct everything or just specifications which can be constructed from authenticated channels. In other words either $\langle \mathfrak{C} \cup \textsc{Auth} \rangle = \langle \{\text{BC}_1\} \rangle = \Omega_1$ or $\langle \mathfrak{C} \cup \textsc{Auth} \rangle = \langle \textsc{Auth} \rangle$.*

## V. Discussion and Open Problems

In this work we have proposed a classification of consistency specifications according to the consistency guarantees they allow to achieve. As a motivating example we have given a complete classification of specifications where a single party can give a binary input. Although we only considered a simple case, the classification provides some unexpected insights into the structure of consistency specifications.


## Acknowledgments

Our research was supported by the Swiss National Science Foundation (SNF), project no. 200020-132794.


## References


[BGP89] Piotr Berman, Juan A. Garay, and Kenneth J. Perry. Towards optimal distributed consensus (extended abstract). In *30th FOCS*, pages 410–415, Research Triangle Park, North Carolina, October 30 – November 1, 1989. IEEE Computer Society Press.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*, pages 136–145, Las Vegas, Nevada, USA, October 14–17, 2001. IEEE Computer Society Press.

[FLM85] 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 PODC*, pages 59–70, Minaki, Ontario, Canada, August 5–7, 1985. ACM.

[FM00] Matthias Fitzi and Ueli M. Maurer. From partial consistency to global broadcast. In *32nd ACM STOC*, pages 494–503, Portland, Oregon, USA, May 21–23, 2000. ACM Press.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. 4:382–401, July 1982.

[Mau04] Ueli Maurer. Towards a theory of consistency primitives. In R. Guerraoui, editor, *International Symposium on Distributed Computing — DISC 2004*, volume 3274 of *Lecture Notes in Computer Science*, pages 379–389. Springer-Verlag, October 2004.

[Mau11] Ueli Maurer. Constructive cryptography – a new paradigm for security definitions and proofs. In S. Moedersheim and C. Palamidessi, editors, *Theory of Security and Applications (TOSCA 2011)*, volume 6993 of *Lecture Notes in Computer Science*, pages 33–56. Springer-Verlag, April 2011.

[MR11] Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *ICS 2011*, pages 1–21, Tsinghua University, Beijing, China, January 7–9, 2011. Tsinghua University Press.

[PW96] Birgit Pfitzmann and Michael Waidner. Information-theoretic pseudosignatures and byzantine agreement for t ≥ n/3. In *Research report*. IBM Research, 1996.