Extended Validity and Consistency in Byzantine Agreement

(Draft)

Matthias Fitzi Martin Hirt Thomas Holenstein

Jürg Wullschleger

Department of Computer Science
Swiss Federal Institute of Technology (ETH)
CH-8092 Zurich, Switzerland
{fitzi,hirt,holenst,wullschleger}@inf.ethz.ch

Abstract

A broadcast protocol allows a sender to distribute a value among a set of players such that it is guaranteed that all players receive the same value (consistency), and if the sender is honest, then all players receive the sender's value (validity). Classical broadcast protocols for n players provide security with respect to a fixed threshold $t < n/3$, where both consistency and validity are guaranteed as long as at most t players are corrupted, and no security at all is guaranteed as soon as $t+1$ players are corrupted. Depending on the environment, validity or consistency may be the more important property.

We generalize the notion of broadcast by introducing an additional threshold $t^+ \geq t$. In a *broadcast protocol with extended validity*, both consistency and validity are achieved when no more than t players are corrupted, and validity is achieved even when up to t^+ players are corrupted. Similarly, we define *broadcast with extended consistency*. We prove that broadcast with extended validity as well as broadcast with extended consistency is achievable if and only if $t + 2t^+ < n$ (or $t = 0$). For example, six players can achieve broadcast when at most one player is corrupted (this result was known to be optimal), but they can even achieve consistency (or validity) when two players are corrupted.

Furthermore, our protocols achieve *detection* in case of failure, i.e., if at most t players are corrupted then broadcast is achieved, and if at most t^+ players are corrupted then broadcast is achieved or every player learns that the protocol failed. This protocol can be employed in the precomputation of a secure multi-party computation protocol, resulting in *detectable multi-party computation*, where up to t corruptions can be tolerated and up to t^+ corruptions can either be tolerated or detected in the precomputation, for any t, t^+ with $t + 2t^+ < n$.

1 Introduction

1.1 Background

Byzantine agreement refers to two slightly different concepts, namely broadcast and consensus. In a *broadcast* protocol, a sender distributes a value among a set of players in such a way that it is guaranteed that all players indeed receive the same value (even when the sender is corrupted). In a *consensus* protocol, a set of players each holding some value decide on one single value, with the property that if they all hold the same value in the beginning then they will decide on this value. A bit more formally, a protocol for Byzantine agreement must satisfy *validity* and *consistency*. Validity means that all players will end up with the correct value in case that the dealer is honest (broadcast), respectively in case that all honest players start with the same input value (consensus). Consistency means that all players must end up with the same value, independent of whether or not the dealer is honest, respectively whether or not all honest players start with the same input value.

The problem of Byzantine agreement was introduced by Lamport, Shostak, and Pease [LSP82]. In a model with pairwise authentic channels, they achieve broadcast (as well as consensus) among n players in presence of an adversary that corrupts up to $t < n/3$ players and make them misbehave arbitrarily. If a secure signature scheme can be used, they achieve broadcast even for any number $t < n$ of corruptions, consensus for up to $t < n/2$. All these bounds are tight [LSP82, KY84, FLM86], but the proposed protocols are inefficient. Efficient protocols were given in [DS83, DFF$^+$82, TPS87, BDDS92, FM97, BGP89, CW92, GM98].

The bounds $t < n$ for broadcast and $t < n/2$ for consensus can also be achieved with unconditional security, when an unconditionally-secure pseudo-signature scheme is set up [BPW91, PW96].

Broadcast is a key ingredient of secure multi-party computation (MPC) protocols. Here, a set of players, each holding a secret input, want to compute an arbitrary function of these inputs in such a way that the inputs remain secret and the outcome of the computation is guaranteed to be correct, even when some of the players are corrupted and misbehave. The problem of MPC was proposed by Yao [Yao82] and first solved by Goldreich, Micali, and Wigderson [GMW87]. This protocol is secure with respect to a computationally bounded adversary that may corrupt up to $t < n/2$ players, which is optimal. When bilateral secure channels are available, security is achievable with respect to an unbounded adversary that corrupts up to $t < n/3$ players [BGW88, CCD88]; also this bound is proven tight. If additionally to the secure channels also secure broadcast channels are available, then information-theoretic security is achievable even for up to $t < n/2$ corruptions [Bea89, RB89, CDD$^+$99].

1.2 Contributions

Classical protocols for Byzantine agreement provide security with respect to a fixed threshold t, where absolute security is guaranteed as long as at most t players are corrupted, and no security at all is guaranteed for the case when $t + 1$ or more players are corrupted. We generalize this notion in the sense that broadcast (according to the classical definition) is achieved as long as up to t players are corrupted, but some (reduced) requirements are still guaranteed even when up to $t^+ \geq t$ players are dishonest.

We propose two concrete primitives:

- In a *broadcast protocol with extended validity*, broadcast is achieved when at most t players are corrupted, and validity (i.e., correctness of the output values) is achieved even when up to t^+ players are corrupted.

- In a *broadcast protocol with extended consistency*, broadcast is achieved when at most t players are corrupted, and consistency (i.e., equality of all outputs) is achieved even when up to t^+ players are corrupted.

For each primitive, we propose an efficient protocol for $t + 2t^+ < n$ (special cases for $t = 0$ are known [FGH$^+$02, GL02] in the literature). Furthermore, the protocol with extended consistency also achieves agreement about the fact whether or not validity is achieved (*validity detection*). The protocol with extended validity can be extended such that even when the sender is malicious, every player receives the same value or learns that no consistency could be reached (*consistency detection*).

As a special case of these results, we can construct protocols for *detectable broadcast*, where broadcast is achieved even when up to t players are corrupted, and either broadcast is achieved or a failure is detected by all honest players when no more than t^+ players are corrupted, for $t + 2t^+ < n$.

This strictly generalizes the result for detectable broadcast in [FGH$^+$02]. This broadcast protocol can be plugged into a multi-party computation protocol for the unconditional model with broadcast (e.g., [Bea89, RB89, CDD$^+$99]), which results in a *detectable MPC protocol* [FGMR02]. Here, the computation is secure and robust as long as at most t players are corrupted, and either the computation is secure or it is aborted before any honest player has distributed his input when up to t^+ players are corrupted, for any t, t^+ with $t + 2t^+ < n$ (respectively, for $t = 0$ and $t^+ < n/2$).
Finally, we prove that the achieved bounds are tight, i.e., broadcast with extended validity (resp. consistency) is impossible for $t + 2t^+ \geq n$.

1.3 Outline

In Section 2, we formally introduce the used model and state some definitions. In Sections 3 and 4, we propose families of efficient deterministic protocols for broadcast with extended validity and extended consistency, respectively. Optimality of our protocols is proven in Section 5, and some final observations and conclusions are given in Section 6.

2 Models and Definitions

We consider a set $P = \{p_1, \ldots, p_n\}$ of players, connected with a complete network of pairwise synchronous authenticated (or secure) channels. The players do not share any consistent information (as a PKI setup). We assume an adaptive adversary that actively corrupts some of the players. The adversary's computational power is unlimited (though the tightness of the protocols will be proved even with respect to a non-adaptive probabilistic polytime adversary).
A broadcast protocol allows a player (the sender) to consistently send a message to all other players, such that all players receive the sender's value, or at least, when the sender is malicious, all players receive the same value.

Definition 1 (Broadcast): Let $P = \{p_1, \ldots, p_n\}$ be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where player $p_s \in P$ (called the *sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ achieves *broadcast* (or is a *broadcast protocol*) with respect to P, p_s, and $\mathcal{D}$, if it satisfies the following conditions:

Validity: If the sender p_s is correct then all correct players p_i decide on the sender's input value, $y_i = x_s$.

Consistency (or Agreement): All correct players decide on the same output value, i.e., if $p_i \in P$ and $p_j \in P$ are correct then $y_i = y_j$.

$\diamond$

In a consensus protocol, every player starts with an input value, and the goal is to make all players agree on the same output value. If all correct players hold the same input value then the output value is required to be the same as this input value.

Definition 2 (Consensus): Let $P = \{p_1, \ldots, p_n\}$ be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where every player $p_i \in P$ holds an input value $x_i \in \mathcal{D}$ and finally decides on an output value $y_i \in \mathcal{D}$ achieves *consensus* (or is a *consensus protocol*) with respect to P and $\mathcal{D}$ if it satisfies the following conditions:

Persistency (or Validity): If all correct players p_i hold the same input value $x_i = v$ then all correct players p_i decide on it, $y_i = v$.

Consistency (or Agreement): All correct players decide on the same output value, i.e., if $p_i \in P$ and $p_j \in P$ are correct then $y_i = y_j$.

◇

When clear from the context, we simply say that a given protocol achieves broadcast (or consensus), neglecting the parameters P, p_s, and $\mathcal{D}$.
Furthermore, we focus on binary Byzantine agreement (domain $\mathcal{D} = \{0, 1\}$) since Byzantine agreement for any finite domain $\mathcal{D}$ can be efficiently solved with any binary protocol [TC84].
Graded consensus, derived from graded broadcast in [FM97], is a variation of consensus where, additionally to the output value, every player gets a grade $g \in \{0, 1, 2\}$ on the outcome of the protocol. We present a slightly modified version with binary grades. If any correct player gets grade 1 then all correct players decide on the same output value, i.e., getting grade 1 implies detecting agreement. If all correct players start with the same input value then all correct players detect agreement, i.e., they get grade 1.

Definition 3 (Graded Consensus): Let $P = \{p_1, \ldots, p_n\}$ be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where every player $p_i \in P$ holds an input value $x_i \in \mathcal{D}$ and finally decides on an output value $y_i \in \mathcal{D}$ and a grade $g_i \in \{0, 1\}$ achieves *graded consensus* with respect to P and $\mathcal{D}$, if it satisfies the following conditions:

Persistency (or Validity): If all correct players p_i hold the same input value $x_i = v$ then all correct players p_i decide on it, $y_i = v$, and get grade $g_i = 1$.

Consistency: If any correct player p_i gets grade $g_i = 1$ then all correct players p_j decide on the same output value $y_i = y_j$.

◇

A broadcast protocol with extended consistency is a protocol which, for two given thresholds t and t^+ with $t \leq t^+$, achieves broadcast as long as no more than t players are corrupted, and achieves consistency (but potentially not validity) as long as no more than t^+ players are corrupted

Definition 4 (Broadcast with extended consistency): Let $P = \{p_1, \ldots, p_n\}$ be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where player $p_s \in P$ (called the *sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ achieves *broadcast with extended consistency* with respect to P, p_s, $\mathcal{D}$, and thresholds t and t^+ if it satisfies the following conditions:

Validity: If the sender p_s is correct and at most t players are corrupted then all correct players p_i decide on the sender's input value, $y_i = x_s$.

Consistency (or Agreement): If at most $f \leq t^+$ players are corrupted then all correct players decide on the same output value.

◇

A broadcast protocol with extended validity is a protocol which, for two given thresholds t and t^+ with $t \leq t^+$, achieves broadcast as long as no more than t players are corrupted, and achieves validity (but potentially not consistency) as long as no more than t^+ players are corrupted

Definition 5 (Broadcast with extended validity): Let $P = \{p_1, \ldots, p_n\}$ be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where player $p_s \in P$ (called the *sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ achieves *broadcast with extended validity* with respect to P, p_s, $\mathcal{D}$, and thresholds t and t^+ if it satisfies the following conditions:

Validity: If the sender p_s is correct and at most t^+ players are corrupted then all correct players p_i decide on the sender's input value, $y_i = x_s$.

Consistency (or Agreement): If at most $f \leq t$ players are corrupted then all correct players decide on the same output value.

$\diamond$

Whereas the computation complexities of all our protocols are obviously polynomial and thus will not be explicitly stated, we state the protocol's communication complexities with respect to two measures. $\mathcal{R}$ denotes the worst-case round complexity, i.e., the maximal possible number of communication rounds required by the protocol. $\mathcal{B}$ denotes the worst-case bit-complexity of a protocol, i.e., the maximal possible number of bits to be sent by all correct players overall during the whole protocol.

3 Broadcast with Extended Validity

We directly present a construction for broadcast with extended validity and consistency detection which is strictly stronger than ordinary broadcast with extended validity.

Definition 6 (Broadcast with Extended Validity and Consistency Detection): Let P be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where player $p_s \in P$ (called the *sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ and a grade value $g_i \in \{0,1\}$ achieves *broadcast with extended validity and consistency detection* (*ECBC*$^+$ for short) with respect to P, p_s, $\mathcal{D}$, and thresholds t and t^+ if it satisfies the following conditions:

Validity: If the sender p_s is correct and at most $f \leq t^+$ players are corrupted then all correct players p_i decide on the sender's input value, $y_i = x_s$.

Consistency: If at most $f \leq t$ players are corrupted then every correct player p_i decides on the same output value $y_i = v$ and $g_i = 1$.

Consistency Detection: If at most $f \leq t^+$ players are corrupted and any correct player p_i computes $g_i = 1$ then every correct player p_j computes $y_j = y_i$.

$\diamond$

Note that however, it is not possible that the players achieve common knowledge about whether or not consistency has been achieved. However, it can be achieved that all players "completely" detect consistency if $f \leq t$ and "soundly" detect consistency if $f \leq t^+$, i.e., $g_i = 1$ always implies reliable detection of consistency.
Note that the special case $t = 0$ (and $t^+ < n$) can be achieved by a protocol wherein the sender simply multi-sends his input to all players who in turn redistribute the received value to everybody (see Protocol `CondGradecast` in [FGH$^+$02] and Protocol 1 in [GL02]). We thus focus on protocols for $t > 0$. The final protocol is based on the implementation of a protocol to solve the following problem:

Definition 7 (Two-level Graded Consensus): A protocol among n players, where every player $p_i \in P$ holds an input value $x_i \in \mathcal{D}$ and every player p_i decides on an value $y_i \in \mathcal{D}$ and a grade $g_i \in \{0,1,2\}$, achieves *two-level graded consensus with respect to thresholds t and t^+* if it satisfies

Persistency: If $f \leq t$ and all correct players p_i enter the protocol with the same input value $x_i = v$ then every correct player p_i computes outputs $y_i = v$ and $g_i = 2$. If $f \leq t^+$ and all correct players p_i enter the protocol with the same input value $x_i = v$ then every correct player p_i computes outputs $y_i = v$ and $g_i \geq 1$.

Consistency: If $f \leq t$ and any correct player p_i computes $g_i \geq 1$ then every correct player p_j computes $y_j = y_i$. If $f \leq t^+$ and any correct player p_i computes $g_i = 2$ then every correct player p_j computes $y_j = y_i$.

$\diamond$

Protocol 1 $\texttt{TLGradedConsensus}_{p_1}(P, x_1, t, t^+)$

```
1.  $\texttt{SendToAll}(x_i)$;  $P$: $\texttt{Receive}(x_i^1, \ldots, x_i^n)$;
2.  $S_i^0 := \Big\{ j \in \{1, \ldots, n\} \mid x_i^j = 0 \Big\}$;  $S_i^1 := \Big\{ j \in \{1, \ldots, n\} \mid x_i^j = 1 \Big\}$;
3.  if $|S_i^{x_i}| \geq n - t^+$ then $z_i := x_i$ else $z_i := \perp$ fi;
4.  $\texttt{SendToAll}(z_i)$;  $P$: $\texttt{Receive}(z_i^1, \ldots, z_i^n)$;
5.  $T_i^0 := \Big\{ j \in \{1, \ldots, n\} \mid z_i^j = 0 \Big\}$;  $T_i^1 := \Big\{ j \in \{1, \ldots, n\} \mid z_i^j = 1 \Big\}$;
6.  if $|T_i^0| \geq |T_i^1|$ then $y_i := 0$ else $y_i := 1$ fi;
7.  if $|T_i^{y_i}| \geq n - t$ then $g_i := 2$
8.  elseif $|T_i^{y_i}| \geq n - t +$ then $g_i := 1$
9.  else $g_i := 0$ fi;
10. return $(y_i, g_i)$;
```

Lemma 3.1. *In Model $\mathcal{M}_{\text{aut}}$, Protocol 1 achieves TLGC if $t + 2t^+ < n$ and $t^+ \geq t$.*

Proof.
Persistency: Suppose that all correct players p_i enter the protocol with the same input value $x_i = v$ and suppose that at most $f \leq t^+$ players are corrupted. Then at least $n - t^+$ correct players distribute value $x_i = v$ in Step 1, and every correct player p_i computes S_i^v such that $|S_i^v| \geq n - t^+$. Furthermore, since $t + 2t^+ < n$, it holds that $|S_i^{1-v}| \leq t^+ < n - t^+$, and every correct p_i computes $z_i = v$ in Step 3. Hence, in Step 4, every such p_i redistributes value $z_i = v$, gets $|T_i^v| \geq n - t^+ > t^+$, and computes $y_i = v$ and $g_i \geq 1$. Finally, if only $f \leq t$ players are corrupted then $|T_i^v| \geq n - t$, and every correct player p_i computes $g_i = 2$.

Consistency: For $v \in \{0, 1\}$, let S_*^v and T_*^v be the set of correct players sending value v in Step 1, and Step 4, respectively. Furthermore, let $F \subset P$ be the set of corrupted players.
Suppose first, that $f \leq t$ players are corrupted ($|F| \leq t$) and that some correct player p_i computes $g_i \geq 1$ and $y_i = v \in \{0, 1\}$. Hence, $|T_i^v| \geq n - t^+$, and since $|F| \leq t$, it follows that $|T_*^v| \geq n - t^+ - t$. Furthermore, as follows from Step 3 of the protocol, for every correct player p_i with $z_i \neq x_i$, it holds that $z_i = \perp$, and hence that $|T_*^v| \leq |S_*^v|$. Therefore, $|T_*^v| \geq n - t^+ - t$ implies for every correct player p_j that $|S_j^v| \geq |S_*^v| \geq |T_*^v| \geq n - t^+ - t$. Additionally, the bound $t + 2t^+ < n$ implies that $|S_j^{1-v}| \leq n - |S_j^v| \leq t^+ + t < n - t^+$, and hence, considering Step 3, that no correct player p_j distributed value $z_j = 1 - v$ during Step 4, i.e., $T_*^{1-v} = \emptyset$. Thus, we get that every correct player p_j computes sets T_j^v and T_j^{1-v} such that $|T_j^{1-v}| \leq |F| \leq t$ and $|T_j^v| \geq |T_*^v| \geq n - t^+ - t > t^+ \geq t \geq |T_j^{1-v}|$, and computes $y_j = y_i$.
Suppose now, that $f \leq t^+$ players are corrupted and that some correct player p_i computes $g_i = 2$. Hence, $|T_i^v| \geq n - t$, and since $|F| \leq t^+$, it follows that $|T_*^v| \geq n - t - t^+$. As above, for the case that $f \leq t$, this implies that $T_*^{1-v} = \emptyset$ (Step 3), and thus every correct player p_j computes sets T_j^v and T_j^{1-v} such that $|T_j^{1-v}| \leq |F| \leq t^+$ and $|T_j^v| \geq n - t^+ - t > |T_j^{1-v}|$, and computes $y_j = y_i$. □

Protocol 2 $\texttt{ExtConsBC}^+{}_{p_1}\,(P, x_1, t, t^+)$

```
1.  $y_i := x_i$;  $h_i := 0$;
2.  for $k := 1$ to $t + 1$ do
3.     if $i = k$ then $\texttt{SendToAll}(y_i)$ fi;    $P$: $\texttt{Receive}(y_i^k)$;
4.     if $h_i = 0$ then $y_i := y_i^k$ fi;
5.     $(y_i, h_i) := \texttt{TLGradedConsensus}(P, y_i, t, t^+)$;
6.  od;
7.  if $h_i = 2$ then $g_i := 1$ else $g_i := 0$ fi;
8.  return $(y_i, g_i)$;
```

Lemma 3.2. *Consider Protocol 2 in Model $\mathcal{M}_{\text{aut}}$, and assume that $t + 2t^+ < n$ and $t^+ \geq t$. Then, for any $k = 2, \ldots, t + 1$, the following holds:*

1. *If at most $f \leq t^+$ players are corrupted and all correct players p_i start Loop k in Step 2 with the same value $y_i = v$ then every correct player p_i holds values $y_i = v$ and $h_i \geq 1$ at the end of the same loop. Additionally, if only $f \leq t$ players are corrupted then $h_i = 2$ at the end of the loop.*
2. *If at most $f \leq t$ players are corrupted and player p_k is correct then, at the end of the loop, every correct player p_i holds the same value $y_i = y_k$ and grade $h_i = 2$.*

Proof.

1. Suppose that $f \leq t^+$ and that every correct player p_i holds value $y_i = v$ at the beginning of the loop. Then, by the persistency property of TLGC, they all compute $y_i = v$ and $h_i \geq 1$ in Step 5, and if $f \leq t$ then even $h_i = 2$ holds.
2. Suppose that $f \leq t$ and that p_k is correct and thus distributes the same value $y_k \in \{0, 1\}$ to all players in Step 3. If every correct player p_i holds grade value $h_i = 0$ then they all enter Protocol `TLGradedConsensus` with $y_i = y_k$, and by its persistency property, compute outputs $y_i = y_k$ and $h_i = 2$. Especially, this holds for $k = 1$ since the players start with grade $h_i = 0$. On the other hand, if any correct player p_i holds $h_i \geq 0$ then, by the consistency property of TLGC which has been priorly invoked, every correct player p_i already holds the same value $y_i = y_k$ before Step 3, and nothing changes until the end of the loop.

□

Theorem 1. *In Model $\mathcal{M}_{\text{aut}}$, Protocol 2 achieves efficient, perfectly secure broadcast with extended validity and consistency detection with sender p_1 if $t + 2t^+ < n$ and $t^+ \geq t$. The round and bit complexities are $\mathcal{R} = 3t + 3$ and $\mathcal{B} = O(n^3)$.*

Proof.
Validity: Suppose that the sender p_1 is correct and that at most $f \leq t^+$ players are corrupted. Then, by Lemma 3.2, after Step 5 of the first loop ($k = 1$), every correct player p_i holds values $y_i = x_1$ and $h_i \geq 1$. Since $h_i \geq 1$ and by the persistency property of Protocol `TLGradedConsensus`, no further loop ($k > 1$) can influence the values y_i and h_i, and every correct player p_i holds value $y_i = x_1$ at the end of the protocol.

Consistency: If $f \leq t$ players are corrupted then there is a player $p_c \in \{p_1, \ldots, p_{t+1}\}$ that is correct. By Lemma 3.2 (2), at the end of loop $k = c$, every correct player p_i holds the same value $y_i = y_c$ and grade $h_1 = 2$. By Lemma 3.2 (1), these values stay persistent until the end of the protocol, and consistency follows.

Consistency Detection: Suppose that at most $f \leq t^+$ players are corrupted and that some correct player p_i computes $g_i = 1$ at the end of the protocol. This implies that $h_i = 2$ after the last invocation of Protocol `TLGradedConsensus`, and by the consistency property of $TLGC$, that every correct player p_j computed $y_j = y_i$ during this invocation and thus exited the protocol with $y_j = y_i$.

The stated complexities can be easily verified by code inspection. □

Note that there is a protocol for broadcast with extended validity *without consistency detection* that requires the same bit complexity but 2 less rounds of communication.

4 Broadcast with Extended Consistency

We directly present a construction for broadcast with extended consistency and validity detection which is strictly stronger than ordinary broadcast with extended consistency.

In contrast to the inherently non-common consistency detection in ECBC$^+$, here it is possible that the players achieve common knowledge about whether or not validity has been achieved.

Definition 8 (Broadcast with Extended Consistency and Validity Detection): Let P be a set of n players and let $\mathcal{D}$ be a finite domain. A protocol Π among P where player $p_s \in P$ (called the *sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ and a grade value $g_i \in \{0,1\}$ achieves *broadcast with extended consistency and validity detection* (*EVBC*$^+$, for short) with respect to P, p_s, $\mathcal{D}$, and thresholds t and t^+ if it satisfies the following conditions:

Consistency: If at most $f \leq t^+$ players are corrupted then every correct player p_i decides on the same pair of output (y, g), $y_i = y$ and $g_i = g$.

Validity: If the sender p_s is correct and at most $f \leq t$ players are corrupted then all correct players p_i decide on the sender's input value, $y_i = x_s$, and grade $g_i = 1$.

Validity Detection: If the sender p_s is correct, at most $f \leq t^+$ players are corrupted, and any correct player p_i computes $g_i = 1$ then every correct player p_j computes $y_j = x_s$. $\diamond$

For didactic reasons, we first sketch a simple protocol for a model with authenticated channels that only guarantees computational security. The protocol for the standard model with secure channels providing unconditional security is stated more explicitly,
Since, for the special case that $t = 0$, efficient and optimally resilient protocols were already given in [FGH$^+$02], we focus on protocols for $t > 0$.

4.1 A Simple Protocol for Computational Security

Protocol 3 `ExtValBC`$^+{}_{p_1}$ (P, x_1, t, t^+)

```
1. Generate a secret-key/public-key pair $(\mathrm{SK}_i, \mathrm{PK}_i)$ according to the key generation algorithm
   of a digital signature system. For every player $p_j \in P$ as a sender, invoke Protocol 2:
   ExtConsBC$^+{}_{p_j}$ $(P, PK_j, t, t^+)$ where $p_j$ inputs his public key $\mathrm{PK}_j$. Store all received public keys
   $\mathrm{PK}_i^1, \ldots, \mathrm{PK}_i^n$ and grades $g_i^1, \ldots, g_i^n$ from these $n$ invocations.
2. $G_i := \bigwedge_{k=1}^{n} g_i^k$.
3. SendToAll($G_i$);  Receive($G_i^1, \ldots, G_i^n$);
   For every player $p_j \in P$ as a sender, an instance of Dolev-Strong broadcast is invoked where $p_j$
   inputs $G_j$. Store all received values as $\Gamma_i^1, \ldots, \Gamma_i^n$.
5. if $|\{j \mid G_i^j = 1\}| > t^+ \ \wedge \ |\{j \mid \Gamma_i^j = 1\}| \geq n - t$ then $g_i := 1$ else $g_i := 0$ fi;
6. If $g_i = 1$ then an instance of Dolev-Strong broadcast is invoked where $p_1$ inputs $x_1$, and its output
   $y_i$ is returned; else $y_i := 0$ is computed.
```

Theorem 2. *In Model $\mathcal{M}_{\mathrm{aut}}$, Protocol 3 achieves broadcast with extended consistency and validity detection with sender p_1 if $t + 2t^+ < n$ and $t^+ \geq t$ as secure as the underlying signature scheme.*
Its round complexity is $\mathcal{R} = 3t + t^+ + 4$ and its bit complexity is polynomial in n, k, and $\log |\mathcal{D}|$ where $\mathcal{D}$ is the domain of the value to be distributed and k is the maximal length of a signature.

Proof.
Consistency: Suppose that $f \leq t^+$ players are corrupted. If every correct player p_i rejects by computing $g_i = 0$ then consistency is satisfied since they all compute $y_i = 0$.
Thus, suppose that some correct player p_i accepts by computing $g_i = 1$. Then $|\{j \mid G_i^j = 1\}| > t^+$, implying that at least one correct player p_k sent $G_k = 1$. Hence, by the definition of ECBC$^+$, all invocations of Protocol 2 achieved validity and consistency (i.e., broadcast when neglecting the grade outputs) implying that all correct players hold each other's authentic public keys. Hence, the invocations of Dolev-Strong broadcast in Step 3 all achieve broadcast and all correct players p_j compute

the same set of values $\Gamma_j^1, \ldots, \Gamma_j^n$. Furthermore, since $g_i = 1$, for every correct player p_ℓ it holds that $|\{j \mid \Gamma_\ell^j\}| \geq n - t$ and thus that $|\{j \mid G_\ell^j = 1\}| \geq n - t - t^+ > t^+$, and all players p_ℓ compute $g_\ell = 1$. Finally, all correct players invoke Dolev-Strong broadcast which now is indeed guaranteed to achieve broadcast, and consistency follows.

Validity: Suppose that $f \leq t$ players are corrupted and that the sender p_1 is correct. Hence, by the definition of ECBC$^+$, all invocations of Protocol 2 achieve validity and consistency (i.e., broadcast when neglecting the grade outputs) and that every correct player p_i computes $g_i = 1$. Thus, all correct players p_i compute $G_i = 1$, all invocations of Dolev-Strong broadcast achieve broadcast, and, in Step 3, the players p_i in turn compute values G_i^j and Γ_i^j such that $|\{j \mid G_i^j = 1\}| \geq n - t > t^+$ and $|\{j \mid \Gamma_i^j = 1\}| \geq n - t$. Finally, all correct players p_i compute $g_i = 1$ and compute $y_i = x_1$ in Step 6.

Validity Detection: Suppose that $f \leq t^+$. We already showed when proving consistency, that if one correct player p_i computes $g_i = 1$, then all correct players hold each other's authentic public keys and all players invoke the Dolev-Strong broadcast protocol. Hence, if the sender p_s is honest the players will indeed compute his input value x_s, according to the properties of Dolev-Strong broadcast.

By inspection of Protocol 2, Step 1 requires $3(t+1)$ rounds. Dolev-Strong broadcast (which is executed in parallel to the multi-send of Step 3, and once again in Step 6) requires another $t^+ + 1$ rounds, and hence the stated round complexity follows. Futhermore, the bit complexities of Protocol 2 and Dolev-Strong broadcast are clearly polynomial in n, k, and $\log|\mathcal{D}|$.

□

4.2 Unconditional Security

We demonstrate the achievability with respect to unconditional security by modifying the Pfitzmann-Waidner precomputation protocol to tolerate $t < n$. However, more efficient solutions can be achieved by modifying the precomputation protocol in [BPW91] to tolerate $t < n/2$. This is possible since $t + 2t^+ < n$ and $t \leq t^+$. However, the Pfitzmann-Waidner protocol is more generic in that it allows for any later broadcast protocol using authentication.

Protocol 4 `ExtValBC`$^+{}_{p_1}$ (P, x_1, t, t^+)

```
1. Execute precomputation the Pfitzmann-Waidner protocol for $b+1$ future broadcasts wherein each
   invocation of broadcast is replaced by an invocation of ECBC$^+$ Protocol 2 with the same sender:
   ExtConsBC$^+{}_{p_k}$ $(P, \cdot, t, t^+)$. Of these instances, one is computed with respect to the intended sender
   $s \in \{1, \ldots, n\}$ of the future broadcast. Of the other $n$ instances, one is computed with respect to
   each player $p_j \in P$.
2. $G_i := \bigwedge_{k=1}^{\ell} g_i^k$ where the $g_i^k$ are all grades received during an invocation of ECBC$^+$ during Step 1.
   Synchronize: Wait and start executing the next step at round $\lfloor \frac{n^2(9t+10)}{2} \rfloor + 1$.
3. SendToAll($G_i$);  Receive($G_i^1, \ldots, G_i^n$);
   For every player $p_j \in P$ as a sender, an instance of Dolev-Strong broadcast is invoked (using
   pseudo-signatures) where $p_j$ inputs $G_j$. Store all received values as $\Gamma_i^1, \ldots, \Gamma_i^n$.
5. if $|\{j \mid G_i^j = 1\}| > t^+ \wedge |\{j \mid \Gamma_i^j = 1\}| \geq n - t$ then $g_i := 1$ else $g_i := 0$ fi;
6. If $g_i = 1$ then an instance of Dolev-Strong broadcast is invoked where $p_1$ inputs $x_1$, and its output
   $y_i$ is returned; else $y_i := 0$ is computed.
```

Theorem 3. *In Model $\mathcal{M}_{\text{sec}}$, for any security parameter $k > 0$, Protocol 4 achieves unconditionally secure broadcast with extended consistency and validity detection (detectable broadcast) with sender p_1 if $t + 2t^+ < n$ and $t+ \geq t$. Thereby the error probability is $\varepsilon < 2^{-k}$.*
Its round complexity is $\lfloor \frac{n^2(9t+10)}{2} \rfloor + 2t^+ + 2$ and its bit complexity is polynomial in n, k, and $\log|\mathcal{D}|$ where $\mathcal{D}$ is the domain of the value to be distributed.

Proof. Consistency, validity, and validity detection follow along the lines of the proof of Theorem 2.

As follows from the analysis in [PW96] and code inspection of Protocol 2, replacing each invocation of broadcast in the Pfitzmann-Waidner precomputation protocol by an invocation of ECBC$^+$ leads to a round complexity of Steps 1 and 2 of at most $\lfloor\frac{n^2(9t+10)}{2}\rfloor$ until all players have finished – note that beyond $f \geq t$, no fault-localization is required but only fault-detection. Steps 3 and 6 each require another $t^+ + 1$ rounds, and hence the stated round complexity follows. Futhermore, the Protocols 2 and Dolev-Strong broadcast are all polynomial in n, k, and $\log|\mathcal{D}|$.

□

5 Impossibility Result

Whereas the case $t = 0$ is obviously are optimal for both, broadcast with extended validity and broadcast with extended consistency, it still needs to be proven that the bound $t + 2t^+ < n$ is optimal. Note that this impossibility result even holds for the ordinary variants without consistency detection, or validity detection, respectively. The proof proceeds along the lines of the impossibility proof in [FLM86] that broadcast is impossible if $t \geq n/3$.

Theorem 4. *In Models $\mathcal{M}_{\text{aut}}$ and $\mathcal{M}_{\text{sec}}$, neither broadcast with extended validity nor broadcast with extended consistency is achievable among a set of n players P if $t > 0$ and $t + 2t^+ \geq n$. For every protocol there exists a value $x_0 \in \{0, 1\}$ such that, when the sender holds input x_0, the adversary can make the protocol fail*

- *with a probability of at least $\frac{1}{6}$ if he is computationally bounded, and*
- *with a probability of at least $\frac{1}{3}$ if he is computationally unbounded.*

The proof of this theorem was moved to the appendix.

6 Conclusions

We have introduced a generalization of broadcast, where either validity (resp. consistency) can be achieved even when more than a third of the players are corrupted, at the costs that consistency (resp. validity) can be guaranteed only when less than a third of the players is corrupted. Such protocols achieve broadcast in the classical sense when up to t players are corrupted, and some reduced notion of broadcast when up to $t^+ \geq t$ players are corrupted, where t^+ can be strictly greater than the number of corruptions tolerable in classical broadcast protocols. The presented protocols are efficient. This extended notion of broadcast has implications in practice, both when broadcast is used as a stand-alone protocol, as well as when it is used as a sub-protocol of some other distributed protocol. For example, it is known that unconditionally-secure multi-party computation robust against $t < n/2$ corruptions is achievable if during a precomputation phase broadcast channels are available. Using broadcast with extended consistency and validity detection, in a model with secure channels but without broadcast, one can fix two parameters t and t^+ with $t + 2t^+ < n$ and start to repeat a precomputation. As soon as the precomputation succeeds (which is guaranteed when at most t players are corrupted) then broadcast will be available unconditionally secure against any number of players, and hence also multi-party computation secure against faulty minorities. In case the protocol does not succeed, all players commonly abort even before having entered any private input to the computation.

References

[BDDS92] A. Bar-Noy, D. Dolev, C. Dwork, and H. R. Strong. Shifting gears: Changing algorithms on the fly to expedite Byzantine agreement. *Information and Computation*, 97(2):205–233, Apr. 1992.

[Bea89] D. Beaver. Multiparty protocols tolerating half faulty processors. In G. Brassard, editor, *Advances in Cryptology—CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pp. 560–572. Springer-Verlag, 1990, 20–24 Aug. 1989.

[BGP89] P. Berman, J. A. Garay, and K. J. Perry. Towards optimal distributed consensus (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pp. 410–415, Research Triangle Park, North Carolina, 30 Oct.–1 Nov. 1989. IEEE.

[BGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pp. 1–10, 1988.

[BPW91] B. Baum-Waidner, B. Pfitzmann, and M. Waidner. Unconditional Byzantine agreement with good majority. In *8th Annual Symposium on Theoretical Aspects of Computer Science*, volume 480 of *lncs*, pp. 285–295, Hamburg, Germany, 14–16 Feb. 1991. Springer.

[CCD88] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pp. 11–19, 1988.

[CDD$^+$99] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations secure against an adaptive adversary. In *Advances in Cryptology — EUROCRYPT '99*, Lecture Notes in Computer Science, 1999.

[CW92] B. A. Coan and J. L. Welch. Modular construction of a Byzantine agreement protocol with optimal message bit complexity. *Information and Computation*, 97(1):61–85, Mar. 1992.

[DFF$^+$82] D. Dolev, M. J. Fischer, R. Fowler, N. A. Lynch, and H. R. Strong. An efficient algorithm for Byzantine agreement without authentication. *Information and Control*, 52(3):257–274, Mar. 1982.

[DS83] D. Dolev and H. R. Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

[FGH$^+$02] M. Fitzi, D. Gottesman, M. Hirt, T. Holenstein, and A. Smith. Byzantine agreement secure against faulty majorities from scratch. Manuscript, 2002.

[FGM01] M. Fitzi, N. Gisin, and U. Maurer. Quantum solution to the Byzantine agreement problem. *Physical Review Letters*, 87(21), Nov. 2001.

[FGMR02] M. Fitzi, N. Gisin, U. Maurer, and O. von Rotz. Unconditional Byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In *Advances in Cryptology — EUROCRYPT 2002*, Lecture Notes in Computer Science, 2002.

[FLM86] M. J. Fischer, N. A. Lynch, and M. Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Computing*, 1:26–39, 1986.

[FM97] P. Feldman and S. Micali. An optimal probabilistic protocol for synchronous Byzantine agreement. *SIAM Journal on Computing*, 26(4):873–933, Aug. 1997.

[GL02] S. Goldwasser and Y. Lindell. Secure computation without a broadcast channel. http://eprint.iacr.org/2002/040.ps, Apr. 2002.

[GM98] J. A. Garay and Y. Moses. Fully polynomial Byzantine agreement for $n > 3t$ processors in $t + 1$ rounds. *SIAM Journal on Computing*, 27(1):247–290, Feb. 1998.

[GMR88] S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. 19th ACM Symposium on the Theory of Computing (STOC)*, pp. 218–229, 1987.

[HH93a] V. Hadzilacos and J. Y. Halpern. The failure discovery problem. *Mathematical Systems Theory*, 26(1):103–129, 1993.

[HH93b] V. Hadzilacos and J. Y. Halpern. Message-optimal protocols for Byzantine agreement. *Mathematical Systems Theory*, 26(1):41–102, 1993.

[KY84] A. Karlin and A. C. Yao. Manuscript, 1984.

[Lam83] L. Lamport. The weak Byzantine generals problem. *Journal of the ACM*, 30(3):668–676, July 1983.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.

[PW96] B. Pfitzmann and M. Waidner. Information-theoretic pseudosignatures and Byzantine agreement for $t >= n/3$. Research Report RZ 2882 (#90830), IBM Research, Nov. 1996.

[RB89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. 21st ACM Symposium on the Theory of Computing (STOC)*, pp. 73–85, 1989.

[TC84] R. Turpin and B. A. Coan. Extending binary Byzantine Agreement to multivalued Byzantine Agreement. *Information Processing Letters*, 18(2):73–76, Feb. 1984.

[TPS87] S. Toueg, K. J. Perry, and T. K. Srikanth. Fast distributed agreement. *SIAM Journal on Computing*, 16(3):445–457, June 1987.

[WC81] M. N. Wegmann and J. L. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, pp. 265–279, 1981.

[Yao82] A. C. Yao. Protocols for secure computations. In *Proc. 23rd IEEE Symposium on the Foundations of Computer Science (FOCS)*, pp. 160–164. IEEE, 1982.

A Impossibility Proof

The following is the proof of Theorem 4.

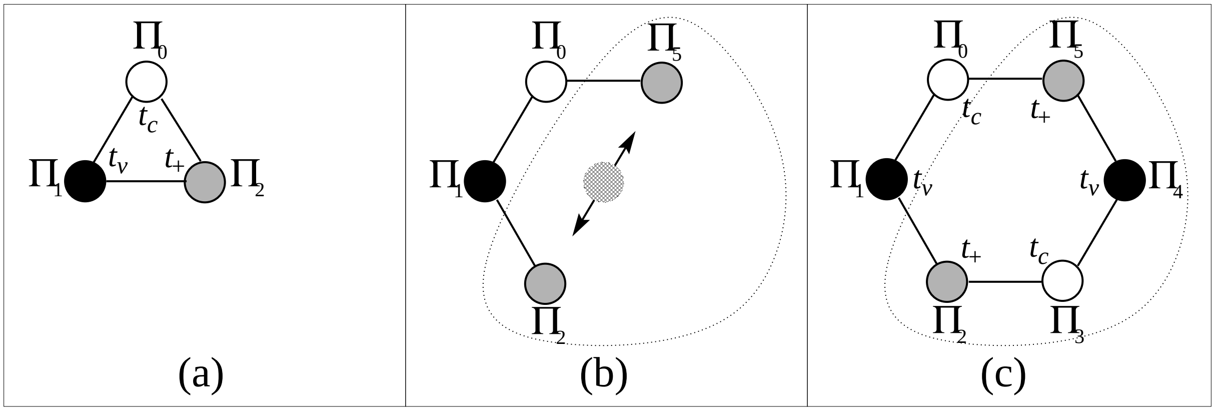

Figure 1: *Rearrangement of processors in the proof of Theorem 4*

Proof. Assume Ψ to be a protocol among n players, $P = \{p_0, \ldots, p_{n-1}\}$, that achieves broadcast with either extended validity or extended consistency for $t > 0$ and $t + 2t^+ \geq n$; and assume p_0 to be the sender with input x_0.

Let $t_c \in \{t, t^+\}$ denote the threshold such that the consistency condition of broadcast is always satisfied when $f \leq t_c$ players are corrupted, and $t_v \in \{t, t^+\}$ denote the threshold such that the validity condition of broadcast is always satisfied when $f \leq t_v$ players are corrupted. In particular $t_v = t^+$ and $t_c = t$ for broadcast with extended validity; and $t_c = t^+$ and $t_v = t$ for broadcast with extended consistency.

Let $\Pi = \{\pi_0, \ldots, \pi_{n-1}\}$ be the set of the players' corresponding processors with their local programs. As follows from the impossibility of standard broadcast, the assumed achievability implies that $t < n/3$, and thus, that $t^+ \geq n/3$. Hence, it is possible to partition the processors into three non-empty sets, $\Pi_0 \dot{\cup} \Pi_1 \dot{\cup} \Pi_2 = \Pi$, such that $1 \leq |\Pi_0| \leq t_c$, $1 \leq |\Pi_1| \leq t_v$, and hence $1 \leq |\Pi_2| \leq t^+$. Note that, hence, $|\Pi_0 \cup \Pi_1| \geq n - t^+$, $|\Pi_1 \cup \Pi_2| \geq n - t_c$, and $|\Pi_2 \cup \Pi_0| \geq n - t_v$.

Furthermore, for each $i \in \{0, \ldots, n-1\}$, let π_{i+n} be an identical copy of processor π_i. For every π_i $(0 \leq i \leq 2n-1)$ let the *type* of processor π_i be defined as the number $i \bmod n$. Finally, for each $k \in \{0, 1, 2\}$, let $\Pi_{k+3} = \{\pi_{i+n} \mid \pi_i \in \Pi_k\}$ form identical copies of the sets Π_k.

Instead of connecting the original processors as required for the broadcast setting, we build a network involving all $2n$ processors (i.e., the original ones together with their copies) by arranging the six processor sets Π_k in a circle. In particular, for all sets Π_k $(0 \leq k \leq 5)$, every processor $\pi_i \in \Pi_k$ is connected (exactly) by one channel with all processors in $\Pi_k \setminus \{\pi_i\}$, $\Pi_{(k-1) \bmod 6}$, and $\Pi_{(k+1) \bmod 6}$. Hence, each processor π_i in the new system is symmetrically connected with exactly one processor of each type (different from his own one) as in the original system. We say that Π_k and Π_ℓ are *adjacent processor sets* if and only if $\ell \equiv k \pm 1 \pmod 6$.

Now, along the lines of [FLM86], for every set $\Pi_k \cup \Pi_{(k+1) \bmod 6}$ $(0 \leq k \leq 5)$ in the new system and without the presence of an adversary, their common view is indistinguishable from their view as the set of processors $\Pi_{k \bmod 3} \cup \Pi_{(k+1) \bmod 3}$ in the original system with respect to an adversary who corrupts all (up to either t or t^+) processors of the remaining processor set $\Pi_{(k+2) \bmod 3}$ in an admissible way.

Let now π_0 and π_n be initialized with different inputs. We now argue that, for each run of the new system, there are at least two pairs $\Pi_k \cup \Pi_{(k+1) \bmod 6}$ $(0 \leq k \leq 5)$ such that the conditions of two-threshold broadcast are not satisfied for them:

By the validity property with respect to t_v, the at least $n - t_v$ players p_i in $\Pi_5 \cup \Pi_0$ must compute $y_i = x_0$ whereas the at least $n - t_v$ players p_i in $\Pi_2 \cup \Pi_3$ must compute $y_i = x_n = 1 - x_0$.

By the consistency property, the at least $n - t_c$ players p_i in $\Pi_1 \cup \Pi_2$ must compute the same output y_i among themselves, and also the at least $n - t_v$ players in Π_4 and Π_5.

Finally, by either the consistency or validity property with respect to t^+, the at least $n - t^+$ players

p_i in $\Pi_0 \cup \Pi_1$ must compute the same output y_i among themselves (since sender $p_0 \in \Pi_0$), and also the at least $n - t^+$ players p_i in Π_3 and Π_4.

Hence, for any possible run of the new system on inputs x_0 and $x_n = 1 - x_0$ it holds that, chosen a pair $(\Pi_k, \Pi_{(k+1) \bmod 6})$ of adjacent processor sets uniformly at random, the probability that the conditions for broadcast are violated for this pair is at least $\frac{1}{3}$.

In particular, there is a pair $(\Pi_k, \Pi_{(k+1) \bmod 6})$ in the new system such that, over all possible runs on inputs $x_0 = 0$ and $x_n = 1$ the probability that the conditions of broadcast are violated for $(\Pi_k, \Pi_{(k+1) \bmod 6})$ is at least $\frac{1}{3}$.

If the adversary is unbounded, given any protocol Ψ, he can compute such a pair $(\Pi_k, \Pi_{(k+1) \bmod 6})$ and act accordingly by corrupting the processors in $\Pi_{(k+2) \bmod 3}$ in the original system, hence forcing the protocol to fail on input

$$x_0 = \begin{cases} 0 & \text{, if } 0 \in \{k, k+1\} \text{ , and} \\ 1 & \text{, else ,} \end{cases}$$

with a probability of at least $\frac{1}{3}$.

If the adversary is computationally bounded then he can still make the protocol fail with a probability of at least $\frac{1}{6}$. □