From Partial Consistency to Global Broadcast*

Matthias Fitzi Ueli Maurer

Department of Computer Science
Swiss Federal Institute of Technology (ETH), Zurich
CH-8092 Zurich, Switzerland

{fitzi,maurer}@inf.ethz.ch

Abstract

This paper considers unconditionally secure protocols for reliable broadcast among a set of n players, some of which may be corrupted by an active (Byzantine) adversary. In the standard model with a complete, synchronous network of pairwise authentic communication channels among the players, broadcast is achievable if and only if the number of corrupted players is less than $n/3$. We show that, by extending this model only by the existence of a broadcast channel among three players, global broadcast is achievable if and only if the number of corrupted players is less than $n/2$. Moreover, for this an even weaker primitive than broadcast among three players is sufficient. All protocols are efficient.

1 Introduction

Broadcast is a fundamental problem in fault-tolerant distributed computing. With respect to the standard model of a synchronous network of pairwise authentic channels, many protocols have been proposed and a large number of results have been published concerning bounds on fault resilience, complexity, and alternative models of network connectivity. It is an interesting open question to analyze these bounds with respect to slightly more powerful communication models such as the standard model extended by partial broadcast among some subsets of the players.

* Appeared at 2000 ACM Symposium on Theory of Computing, May 2000. Research supported by the Swiss National Science Foundation (SNF), SPP project no. 5003-045293

1.1 Contributions

We consider the general problem of reductions among various types of primitives guaranteeing some form of consistency, in the presence of an adversary who can corrupt certain players. It is well known that the strongest form of consistency, namely consensus or broadcast, can be achieved among a set of n players connected by pairwise authenticated channels if and only if the number t of cheaters is less than $n/3$. The main result of this paper is that broadcast secure against any $t < n/2$ cheaters can be achieved by only assuming an additional primitive satisfying some weak form of consistency that is not realizable for $t < n/3$. One example of such a sufficient primitive is a broadcast channel among three players, but even a weak form of broadcast among three players suffices. Also any broadcast among n_0 players tolerating $\lceil n_0/3 \rceil$ cheaters is sufficient.

1.2 Motivation

There are several motivations for this work. First, we hope to initiate a new line of research on reductions among consistency primitives, by giving a few non-trivial examples. Second, the question of whether the bound $t < n/3$ can be improved is a very natural one. As it has been proved that $t < n/3$ is a tight bound, one must assume some additional primitive more powerful than just authenticated channels, and it is natural to assume the weakest possible primitive not yet implied by the considered model. Third, it is quite possible that some of these primitives exist in nature (e.g., based on exploiting some quantum phenomenon, or simply due to the topology of the communication network), and this would imply that the important broadcast primitive could be realized even for $t < n/2$ instead of only $t < n/3$. Moreover, one can show that the same improvement also applies to the more general task of secure multi-party computation.

1.3 Broadcast

The goal of broadcast among a set of players is to have one specific player, called the dealer, consistently distribute some input value to all the remaining players. Since our model does not assume a physical channel that provides consistency, this functionality must be simulated by a protocol among the players. A broadcast protocol must satisfy the following conditions:

Agreement: All correct players decide on the same output value.

Validity: If the dealer is correct then all correct players decide on the dealer's input value.

Termination: All correct players terminate the protocol after a finite number of communication rounds.

Consensus is a closely related problem, in which every player initially holds his own input value to the protocol. Again, every player must decide on an output value such that the former agreement and termination properties are still satisfied, while the validity condition is replaced by

Persistency: If all (correct) players initially hold the same input value v then all correct players decide on v. In other words, i.e., pre-agreement on a value remains persistent.

1.4 The two-cast model

In this paper we consider a set P of n players. The goal is to achieve broadcast unconditionally secure against an active (Byzantine) threshold adversary that may corrupt up to t of the n players, i.e., the adversary may take full control over the corrupted players and make them deviate from the prescribed protocol in an arbitrary way. *Unconditional security* means that, for some arbitrarily small (but *a priori* fixed) error probability ε, the probability that the protocol achieves broadcast is at least $1 - \varepsilon$ (while the outcome is arbitrary if the protocol fails) whereas no assumptions are made about the adversary's computational power. As a special case of unconditional security, *perfect security* allows no probability of error ($\varepsilon = 0$).
We assume the standard communication model with a complete (fully connected) synchronous network of pairwise authentic channels among the players extended by unconditionally secure, synchronous broadcast channels[1] among each triple of players, i.e., for each subset of three players ($S \subset P$, $|S| = 3$) and for any selection of a dealer among them there is a broadcast channel from the dealer to the remaining two players. Such broadcast channels from a dealer to two receivers will be denoted as *two-cast channels*. The security of the two-cast channels is not necessarily required to be perfectly secure (i.e., to have zero error probability) but we assume their error probability ε_0 to be customizable to an arbitrarily small level. Hence we distinguish the *perfect two-cast model* where the two-cast channels are assumed to be perfectly secure ($\varepsilon_0 = 0$), and the *unconditional two-cast model* where the two-cast channels are allowed to have some negligible error probability $\varepsilon_0 > 0$.

[1] In fact, such a broadcast channel might again be simulated by a synchronous protocol among the involved players, for instance based on a quantum physical phenomenon.

1.5 Previous work

For the standard communication model with a complete synchronous network of pairwise authentic channels, Pease, Shostak, and Lamport [17] proved that perfectly secure broadcast is achievable if and only if less than a third of the players is corrupted: $t < n/3$. This tight bound more generally holds with respect to unconditional security, i.e., when even allowing a negligible error probability, as proven by Karlin and Yao [15]. For the same model numerous unconditionally secure protocols with optimal resilience have been proposed in the literature [9, 1, 19, 10, 3, 5, 13] which all have communication and computation complexities polynomial in the number of players.
The extension of the standard communication model by partial broadcast was already considered by Franklin, Wright, and Yung in [11, 12] in the context of secure point-to-point communication over an incomplete network — a problem initially studied by Dolev, Dwork, Waarts, and Yung [8] for the standard communication model. The problem in [11] is to achieve private point-to-point communication in the presence of a passive adversary, given partial-broadcast but not necessarily private communication channels among pairs of players. [12] considers secure point-to-point communication over local-broadcast networks in the presence of an active adversary.

1.6 Notation

The player set is denoted by $P = \{p_1, \ldots, p_n\}$. Without loss of generality we assume p_1 to be the dealer of the broadcast. All pseudo-code descriptions of protocols are stated with respect to the local view of the player p who stands for any arbitrary player in P. The complete protocols consist of all players executing their local codes in parallel. Variables that have no subscript (e.g. v) are stated with respect to an arbitrary player and variables with a subscript p (e.g. v_p) denote the corresponding variable of the particular player p.
The protocol descriptions do not explicitly describe how to handle received messages that are outside the value domain as expected for the protocol, e.g., if some player expects a value $v \in \{0, 1\}$ from another player but instead receives a value $v \notin \{0, 1\}$. For these cases we always implicitly assume a correct player to substitute the received value by some arbitrary value inside the required domain.
Finally, in our protocol constructions, we focus on achieving broadcast (and consistency primitives in general) where the domain of values is restricted to $\{0, 1\}$ since protocols for any finite domain can be easily obtained from any bit-protocol (e.g., by using the construction in [20]). In fact, the generalization to any finite domain could even be directly achieved by slight modification of the described bit-protocols.

1.7 Outline

Section 2 describes a protocol construction for efficient broadcast among n players in the two-cast model, unconditionally secure against $t < n/2$ actively corrupted players. In Section 3, $t < n/2$ is proven to be a tight bound for the achievability of broadcast. In Section 4, we first prove that even a weaker form of two-cast is sufficient to achieve broadcast among n players in the presence of $t < n/2$ player corruptions, and finally prove a large class of consistency primitives to be equivalent. Implications on general multi-party computation are discussed in Section 5.

2 Efficient broadcast protocol

This section describes a broadcast protocol for n players in the perfect two-cast model that is perfectly secure against an adversary that corrupts any minority $t < n/2$ of the players. At the end of this section we shall see that the same protocol is still unconditionally secure when the underlying two-cast channels involve some negligible error probability (i.e., in the unconditional two-cast model).

2.1 Graded consensus implies broadcast

A common approach to construct broadcast protocols is to find protocols to solve weaker problems, e.g., graded broadcast by Feldman and Micali [10], and then to achieve the strong requirements of broadcast by composing the weak protocols in a clever way. While the constructions in [10] additionally involve common coins, Berman, Garay, and Perry [3] proposed broadcast protocols that only rely on a consensus variant of graded broadcast, which we shall denote by graded consensus, and on the fact that there is at least one correct player. This implies that, whenever $t < n$, the achievability of graded consensus immediately implies the achievability of broadcast as long as at least pairwise communication is possible. Hence, with respect to our model, it is sufficient to give a protocol construction for graded consensus, since this protocol can then be extended along the lines of [3]. How to achieve this extension is described in the following paragraphs.

2.1.1 Graded consensus

Graded consensus is a weak variant of consensus — yet with the same persistency condition but with a weakened agreement property which we shall refer to as its *consistency* property.
Every player enters the protocol with some value $v \in \{0,1\}$ and finally decides on a value $v' \in \{0,1\}$. Moreover, as a further output of the protocol, every player receives a grade value $g \in \{0,1\}$ to be interpreted as a rating on the level of agreement that has been achieved, i.e., $g = 0$ for reject, and $g = 1$ for accept.[2]
While pre-agreement cannot be invalidated by this protocol due to its persistency property, the adversary will still have the power to prevent agreement in any other case. However, an accepting player ($g = 1$) always knows that all correct players decided on the same value v', i.e., he detects agreement. This property will be crucial in order to later extend this protocol to a broadcast protocol.

Definition 1: A protocol achieves *graded consensus* if it satisfies the following conditions.

Consistency: If any correct player p accepts a value $v'_p \in \{0,1\}$ with $g_p = 1$ then, for every correct player q, $v'_q = v'_p$.

Persistency: If all correct players enter the protocol with the same input $v \in \{0,1\}$ then $v'_p = v$ and $g_p = 1$ for every correct player p.

The following theorem is an immediate consequence of Lemmas 1 and 2 in the next sections.

Theorem 1 *If pairwise authentic communication is possible among the n players, then, for any number t of potential player corruptions, the achievability of graded consensus implies the achievability of broadcast. Moreover, efficiency of graded consensus implies efficiency of broadcast.*

2.1.2 King consensus

A variant of graded consensus can be achieved by, after first executing a graded-consensus protocol, making some designated player p_k, called the king [3], redistribute his resulting value of the graded-consensus protocol. Finally, every player who did accept the outcome v' of the graded-consensus protocol ($g = 1$) sticks to this value whereas all other players ($g = 0$) decide on the value received by the king. We refer to this protocol as the `KingConsensus` protocol.

Protocol $\texttt{KingConsensus}_{p_k}(P, v)$**:**
1. $(v, g) := \texttt{GradedConsensus}(P, v)$;
2. if $p = p_k$ then $\texttt{SendToAll}(v)$; $w := v$ else $\texttt{Receive}(w)$ fi;
3. if $g = 1$ then $v' := v$ else $v' := w$ fi;
4. return v';

It is easy to see that this protocol still maintains persistency. Moreover, agreement is even achieved whenever p_k is correct (which, of course, is generally unknown).

Definition 2: A protocol achieves *king consensus* (with respect to p_k) if it satisfies the following conditions.

Consistency: If player p_k is correct then all correct players agree on the same value $v' \in \{0,1\}$ at the end of the protocol.

Persistency: If all correct players enter the protocol with the same input $v \in \{0,1\}$ then $v'_p = v$ for every correct player p.

Lemma 1 *Protocol* `KingConsensus` *achieves king consensus.*

[2] Note that graded broadcast in [10] originally worked with three grade values (reject, semi-accept, and accept). However, the intermediary grade value is not necessary for our construction.

Proof: *Consistency:* Suppose player p_k to be correct. If every correct player p accepts p_k's value by setting $v'_p := w$ then all correct players trivially agree on the same value, since p_k distributed the same value to every other player. On the other hand, suppose that any correct player p ignores p_k's value by setting $v'_p := v_p$ since $g_p = 1$. Since this implies agreement after the execution of graded consensus, especially p_k holds and redistributes this value. Hence, every correct player will decide on this value independently on whether or not he adopts p_k's value.

Persistency: If all correct players enter the protocol with the same input v then, for every correct player p, $v_p = v$ and $g_p = 1$ after the execution of `GradedConsensus`, and hence $v'_p = v$ at the end of the protocol. ∎

2.1.3 Broadcast

Finally, broadcast can be achieved by first having the dealer p_1 distribute his value and then appending t instances of `KingConsensus` with distinct kings $p_k \in P \setminus \{p_1\}$.

Protocol $\texttt{Broadcast}_{p_1}(P, v)$**:**

1. if $p = p_1$ then `SendToAll` (v) else `Receive` (v) fi;
2. for $k := 2$ to $t+1$ do $v := \texttt{KingConsensus}_{p_k}(P, v)$ od;
3. return v;

Lemma 2 *Protocol* `Broadcast` *achieves broadcast if at most t players are corrupted.*

Proof: *Consistency:* If the dealer is correct then agreement on his input holds before the first `KingConsensus` protocol is executed. Hence, by Lemma 1 agreement on the dealer's input will persist until the end of the protocol.

Agreement: If the dealer is corrupted then there is at least one correct player in $\{p_2, \ldots, p_{t+1}\}$ and hence, after this player's `KingConsensus`, agreement holds by Lemma 1.

Termination: Termination is trivially satisfied by construction. ∎

2.2 Achieving graded consensus

This section presents a protocol construction for graded consensus in the two-cast model. The construction proceeds in three steps. In Section 2.2.1, two-cast is extended to a majority-voting protocol among (still) three players. Any invocation of two-cast will always be encapsulated by this protocol, i.e., two-cast will not be used in any other context. Section 2.2.2 shows how to build a weak consensus variant on top of majority-voting among three players, which then in Section 2.2.3 is extended to a graded-consensus protocol.

2.2.1 Triple-majority voting

This section describes the basic sub-protocol that exploits the power of two-cast. For simplicity, let's assume that two-cast works in a way that, besides the two actual receivers, also the sender receives an output which is equal to his input value.
The protocol `MajorityVoting` is defined for any subset of three players $\{q, r, s\} \subset P$ with every player initially holding an input value $v \in \{0, 1, 2\}$, i.e., a value from the original domain extended by an invalidity value 2, and finally deciding on an output value $v' \in \{0, 1, 2\}$.

Protocol $\texttt{MajorityVoting}(\{q, r, s\}, v)$**:** First, q, r, and s two-cast their initial values v_q, v_r, and v_s. Second, every player decides on the majority value among his outputs of the three two-casts, or on 2 if no majority exists. Let v^q, v^r, and v^s be the values that are effectively received by a player $p \in \{q, r, s\}$. Then p decides on

$$v' := \begin{cases} v^x & \text{, if } \exists x, y \in \{q, r, s\}, x \neq y : v^x = v^y, \\ 2 & \text{, else.} \end{cases}$$

Lemma 3 *For any number of corrupted players among $\{q, r, s\}$, all correct players decide on the same output value. If at most one player is corrupted and any two correct players enter the protocol with the same value v, then every correct player finally decides on $v' = v$.*

Proof: The lemma immediately follows from the properties of two-cast and from the construction of the protocol. ∎

The `MajorityVoting` protocol will always be applied for all $\binom{n}{3}$ distinct subsets $\{q, r, s\} \subset P$ of three players in parallel. Thus, during such a round of protocol invocations, every player p receives an output value v^{pqr} for each subset $\{q, r\} \subset P \setminus \{p\}$.[3] Furthermore we assume the player set to be ordered, i.e., for any two players q and r, $(q < r) \Leftrightarrow \neg(r < q)$, such that the expression "$\forall q, r \in P \setminus \{p\}, (q < r)$" quantifies over every subset $\{q, r\} \subset P \setminus \{p\}$ exactly once. Finally, we define

$$v^{pq*} \equiv w \quad :\Longleftrightarrow \quad \forall r \in P \setminus \{p, q\} : v^{pqr} = w$$

to express that all `MajorityVoting` protocols that involve both of the players p and q result in w.

2.2.2 Weak consensus

Weak consensus is a variant of crusader agreement in [7] and satisfies the same conditions as the Makeunique protocol in [14]. It can be seen as an even weaker consensus variant than graded consensus. Every player enters the protocol with some value $v \in \{0, 1\}$ and finally decides on a value $v' \in \{0, 1, 2\}$. Every player will decide on a value $v' \in \{0, 1\}$ if and only if, according to his view, agreement on $v = v'$ could have been satisfied at the beginning of the protocol — otherwise he will decide on $v' = 2$.

[3] Due to the set-based definition, $v^{pqr} = v^{qpr} = \ldots$ all denote the same value for any permutation of the occurring players.

Definition 3: A protocol achieves *weak consensus* if it satisfies the following conditions.

Consistency: If $v'_p \in \{0,1\}$ for any correct player p then $v'_q \in \{v'_p, 2\}$ for every correct player q.

Persistency: If all correct players enter the protocol with the same input $v \in \{0,1\}$ then $v'_p = v$ for every correct player p.

Protocol `WeakConsensus` (P,v)**:**[4]

1. $\forall q,r \in P\setminus\{p\}, (q<r): v^{pqr}:=$ `MajorityVoting` $(\{p,q,r\},v)$;
2. $X^0 := \{\, q \in P\setminus\{p\} \;:\; v^{pq*} \equiv 0 \,\}$;
3. $X^1 := \{\, q \in P\setminus\{p\} \;:\; v^{pq*} \equiv 1 \,\}$;
4. if $|X^0| \geq n-t-1$ then $v' := 0$
5. elseif $|X^1| \geq n-t-1$ then $v' := 1$
6. else $v' := 2$
7. fi;
8. return v'

Lemma 4 *The protocol* `WeakConsensus` *guarantees that, for any correct player p and any value $w \in \{0,1\}$, $|X_p^w| \neq \emptyset$ implies $X_p^{1-w} = \emptyset$.*

Proof: $q \in X_p^w$ implies $v^{pqr} = w$ for all $r \in P\setminus\{p,q\}$ and hence there is no $r \in P\setminus\{p,q\}$ satisfying $v^{prq} = 1-w$ which implies $X_p^{1-w} = \emptyset$. ■

Lemma 5 *For any two correct players p and q and any value $w \in \{0,1\}$ the sets X_p^w and X_q^{1-w} are disjoint: $X_p^w \cap X_q^{1-w} = \emptyset$.*

Proof: If $p = q$ then the lemma immediately follows from Lemma 4. Suppose now that $p \neq q$ and that there is a player $r \in P$ and a value $w \in \{0,1\}$ such that $r \in X_p^w \cap X_q^{1-w}$. Then $w = v_p^{prq} = v_q^{qrp} = 1-w$ in contradiction to the consistency of `MajorityVoting` as stated in Lemma 3. ■

Theorem 2 *Protocol* `WeakConsensus` *achieves weak consensus among $n \geq 3$ players secure against $t < n/2$ actively corrupted players.*

Proof: *Consistency:* For the sake of contradiction, suppose $v'_p = w \in \{0,1\}$ and $v'_q = 1-w$. Then, according to the protocol, $|X_p^w| \geq n-t-1$ and $|X_q^{1-w}| \geq n-t-1$. First note that $q \notin X_p^w$ (and hence by symmetry $p \notin X_q^{1-w}$) since otherwise for any $q_i \in X_q^{1-w}$ we would get $1-w = v^{qq_ip} = v^{pqq_i} = w$ would hold. Since also $X_p^w \cap X_q^{1-w} = \emptyset$ by Lemma 5, the sets X_p^w, X_q^{1-w}, and $\{p,q\}$ are pairwise disjoint and hence $n = |P| \geq |X_p^w \cup X_q^{1-w} \cup \{p,q\}| = |X_p^w| + |X_q^{1-w}| + 2 \geq 2(n-t-1)+2 = 2(n-t) > 2(n-\frac{n}{2}) = n$, which is a contradiction.

Persistency: Let C be the set of correct players. Since all correct players input the same value $v \in \{0,1\}$, $C\setminus\{p\} \subset X_p^v$ for every correct player p and hence $|X_p^v| \geq n-t-1$. By Lemma 4, $X_p^{1-v} = \emptyset$ and hence $v_p = v$ at the end of the protocol. ■

[4]Remember that pseudo-code is stated with respect to the local view of player p (Section 1.6).

2.2.3 Graded consensus

We are now ready to construct a graded-consensus protocol on top of the protocol for weak consensus of the previous section. Refer to the beginning of Section 2.1.1 for the definition of graded consensus.

Protocol `GradedConsensus` (P,v)**:**

1. $v :=$ `WeakConsensus` (P,v);
2. $\forall q,r \in P\setminus\{p\}, (q<r): v^{pqr}:=$ `MajorityVoting` $(\{p,q,r\},v)$;
3. $Y^0 := \Big\{ q \in P\setminus\{p\} \;:\; \big|\{ r \in P\setminus\{p,q\} : v^{pqr} = 0 \}\big| \geq t \Big\}$;
   $Z^0 := \Big\{ q \in P\setminus\{p\} \;:\; v^{pq*} \equiv 0 \Big\}$;
4. $Y^1 := \Big\{ q \in P\setminus\{p\} \;:\; \big|\{ r \in P\setminus\{p,q\} : v^{pqr} = 1 \}\big| \geq t \Big\}$;
   $Z^1 := \Big\{ q \in P\setminus\{p\} \;:\; v^{pq*} \equiv 1 \Big\}$;
5. if $|Y^0| > 0$ then $v' := 0$ else $v' := 1$ fi;
6. if $|Z^{v'}| \geq t$ then $g := 1$ else $g := 0$ fi;
7. return (v', g);

Lemma 6 *If in the protocol* `GradedConsensus`*, for some correct player p and some value $w \in \{0,1\}$, $Y_p^w \neq \emptyset$, then $Y_q^{1-w} = \emptyset$ for every correct player q.*

Proof: Let p and q be two (not necessarily distinct) correct players and for some $w \in \{0,1\}$ and $r \in P\setminus\{p\}$ assume that $r \in Y_p^w$. Hence

$$\exists R = \{r_1,\ldots,r_t\} \subset P\setminus\{p,r\} : \forall r_i \in R : v^{prr_i} = w \;.$$

`MajorityVoting` guarantees that the resulting value equals 2 if all inputs differ. Hence it must hold either that player p had input $v_p = w$ for all protocols `MajorityVoting` $(\{p,r,r_i\},v)$ or that player r and all players $r_i \in R$ had input w for these protocols. Since p and at least one player in $\{r,r_1,\ldots,r_t\}$ are correct, `WeakConsensus` must have resulted in $v' = w$ for at least one correct player.
On the other hand, the same argumentation would hold if $s \in Y_q^{1-w}$ for any $s \in P\setminus\{q\}$, i.e., $Y_q^{1-w} \neq \emptyset$ would imply that `WeakConsensus` must also have resulted in $v' = 1-w$ for at least one correct player, which is impossible by Theorem 2. ■

Theorem 3 *Protocol* `GradedConsensus` *achieves graded consensus among $n \geq 3$ players secure against $t < n/2$ actively corrupted players.*

Proof: *Consistency:* Suppose that some correct player p accepts some $v'_p = w$ with $g_p = 1$. Hence $|Z_p^w| \geq t$, i.e.,

$$\exists R = \{r_1,\ldots,r_t\} \subset P\setminus\{p\} : \forall r_i \in R : v^{pr_i*} \equiv w \;,$$

and for every correct player $q \neq p$ either $q \in R$ and hence $v^{pq*} \equiv v^{qp*} \equiv w$ or $q \notin R$ and $\forall r_i \in R : v^{pr_iq} = v^{qpr_i} = w$, both of which imply $p \in Y_q^w$. Thus $Y_q^w \neq \emptyset$, and by Lemma 6 $Y_q^{1-w} = \emptyset$, and hence $v'_q = w = v'_p$.

Persistency: If all correct players enter `GradedConsensus` with the same value $v = w \in \{0, 1\}$ then, by Theorem 2, all correct players still hold value w after `WeakConsensus` and use it as an input for all `MajorityVoting` protocols they are involved in. Hence for every correct player p $|Z_p^w| \geq t$ since $v^{pq*} \equiv w$ for every other correct player $q \neq p$ (of which there are at least t), and $v'_p = w$ and $g_p = 1$. ■

2.3 Broadcast

Theorem 4 *In the perfect (unconditional) two-cast model, perfectly (unconditionally) secure broadcast among $n \geq 3$ players is achievable if $t < n/2$. Moreover, there exist protocols with communication and computation complexities polynomial in n.*

Proof: Achievability in the perfect model follows from Theorem 1 and Theorem 3. Efficiency can be easily verified by code inspection of the `Broadcast` protocol: $3t + 1$ communication rounds, tn^3 two-cast invocations, $O(tn^3)$ overall message bit complexity, and $O(tn^2)$ local computation per player.
In order to achieve unconditionally secure broadcast in the unconditional model exactly the same protocol can be used. While the broadcast protocol remains perfectly secure if none of the two-casts fails, it must already be considered to fail if any single two-cast invocation fails.[5] Hence, an upper bound on the error probability ε of the broadcast protocol when given the error probability ε_0 of the underlying two-cast, can be estimated as the number of two-cast invocations times ε_0. The protocol involves t rounds of `KingConsensus` each of which involves two rounds of `MajorityVoting` among all $\binom{n}{3}$ sets of three players. Finally, each `MajorityVoting` involves three two-cast invocations. Hence the entire broadcast protocol involves a total number of $6t \cdot \binom{n}{3} < tn^3$ single two-cast invocations, which yields an error probability of $\varepsilon \leq tn^3\varepsilon_0$. Hence, in order to achieve broadcast with error probability at most ε, for some given ε, the error probability ε_0 of the underlying two-cast can be customized to $\varepsilon_0 \leq \frac{\varepsilon}{tn^3}$, i.e., ε reduced by a factor polynomial in n. ■

3 Tightness of the $(n/2)$-bound

In this section we prove that, in the two-cast model, unconditionally secure broadcast is not achievable if at least half of the players are actively corrupted.
Our proof makes use of the ideas in [16] for the impossibility of agreement among three players with one Byzantine fault with respect to the standard model with only pairwise communication channels. The idea there is to suppose that there exists such a protocol involving three processors which then can be used to build a different system with contradictory behavior, hence proving that such a protocol cannot exist. Note that we do not require this new system to solve the broadcast problem, it is just a distributed system whose behavior is determined by the local programs and inputs of the involved processors which can achieve broadcast when being arranged in the original way. Nor is there anymore an adversary to take control of any processor. We will only argue that for some processor pairs that are considered to be correct, in the new system (without the presence of an adversary), their views (while being correct) are indistinguishable from their views in the original system for some particular strategy of an admissible adversary (with respect to the original broadcast), and that hence all conditions for broadcast must still hold with respect to every such a pair of processors.
We first show that broadcast is impossible for the special case of $n = 4$ and $t \geq 2$. The general case can then be shown by a generalization of this proof.

Lemma 7 *Given only pairwise communication channels and two-cast among each triple of players, unconditionally secure broadcast among $n = 4$ players is not achievable if $t \geq 2$.*

Proof: Suppose, for the sake of contradiction, that there is a protocol that achieves broadcast for the four players $p_0, \ldots, p_3$ with p_0 being the dealer, even if up to two of the players are actively corrupted. Let $\pi_0, \ldots, \pi_3$ denote the players' corresponding processors with their local programs and, for each $i \in \{0, \ldots, 3\}$ let π_{i+4} be an identical copy of processor π_i. Instead of connecting the four original processors as prescribed for the setting in which they can be used for broadcast, we build a network among all eight processors (i.e., the original ones together with their copies) in the following way:

In the original system, each processor π_i communicates with the processors π_{i-1}, π_{i+1}, and π_{i+2} (interpreting the indices modulo 4). Instead, the pairwise communication channels are reconnected such that each processor π_i sends his outgoing messages to the processors π_{i-1}, π_{i+1}, and π_{i+2}, interpreting the indices modulo 8 instead of modulo 4.
In the original system, each processor π_i communicates via two-cast with the processor pairs (π_{i-2},π_{i-1}), (π_{i-1},π_{i+1}), and (π_{i+1},π_{i+2}) (again interpreting the indices modulo 4). Instead, the two-cast channels are reconnected such that each processor π_i two-casts his outgoing messages to the processor pairs (π_{i-2},π_{i-1}), (π_{i-1},π_{i+1}), and (π_{i+1},π_{i+2}), interpreting the indices modulo 8.

It is now easy to see that the situation for every pair of adjacent processors π_i and $\pi_{((i+1) \mod 8)}$ is completely consistent with the situation of the two adjacent processors $\pi_{(i \mod 4)}$ and $\pi_{((i+1) \mod 4)}$ in the original system:

- Any message that would have been transferred among π_i and π_{i+1} in the original system is still transferred among them in the new system.

- Any two-cast for the receivers π_i and π_{i+1} in the original system is still addressed to the same processors π_i and π_{i+1} in the new system.

[5] Note that we even allow two-cast among three correct players to fail with the given error probability.

Hence, for every pair of adjacent processors π_i and $\pi_{((i+1) \mod 8)}$, their common view is completely indistinguishable from their view as two processors $\pi_{(i \mod 4)}$ and $\pi_{((i+1) \mod 4)}$ in the original system with respect to an adversary that corrupts the remaining two processors $\pi_{((i+2) \mod 4)}, \pi_{((i+3) \mod 4)}$ in a certain admissible way.

This new system involves two processors of the type corresponding to the dealer, namely π_0 and π_4, that are the only processors that enter an input. Suppose now that π_0 and π_4 have distinct inputs in $\{0,1\}$, i.e., that without loss of generality, π_0 has input $v_0 = 0$ and that π_4 has input $v_4 = 1$.[6]

We now argue that there are at least two pairs of adjacent processors (i.e., one fourth among all eight such pairs) for which the broadcast conditions are not satisfied although being completely consistent with two correct processors in the original system. For this we distinguish two cases:

- **Agreement holds** for all pairs of adjacent processors, i.e., all eight processors decide on the same value $v \in \{0,1\}$. Then both pairs that involve the dealer with input $1 - v$ (either π_0 or π_4) violate the validity property of broadcast.

- **Agreement does not hold** for all pairs of adjacent processors. Then there must be at least two such pairs deciding on distinct values since the processors are arranged in a circle.

Hence there must be some pair of adjacent processors $(\pi_i, \pi_{((i+1) \mod 8)})$ that fails with a probability of at least $\frac{1}{4}$. Otherwise strictly less than two pairs would fail per such invocation of the new system. Let now σ_0 be the probability that a dealer selects input 0. Then over all invocations of the new system (for arbitrary inputs of π_0 and π_4) the same pair still fails with a probability of at least $\frac{1}{4}\sigma_0(1-\sigma_0)$ (i.e., with a probability of at least $\frac{1}{4}$ in all runs where $v_0 = 0$ and $v_4 = 1$). Hence, there is an admissible adversary strategy in the original system of four processors to make the according pair $(\pi_{(i \mod 4)}, \pi_{((i+1) \mod 4)})$ fail with a probability of at least $\frac{1}{4}\sigma_0(1-\sigma_0)$, which is non-negligible. ■

Theorem 5 *Given only pairwise communication channels and broadcast channels among each triple of players, unconditionally secure broadcast among $n > 3$ players is not achievable if $t \geq n/2$.*

Proof: The proof of Lemma 7 can be generalized for any number of players. For simplicity, suppose $n = 2k$ to be even and $t \geq k$ (the case $n = 2k+1$ and $t \geq k+1$ can be easily reduced to the even case by neglecting one of the players completely, which can be interpreted as a special kind of active corruption of this particular player, hence reducing this case to $n = 2k$ and $t \geq k$).

For each $i \in \{0, \ldots, n-1\}$ let π_{i+n} again be an identical copy of processor π_i. The resulting set of $2n$ processors is partitioned into eight blocks $\Pi_0, \ldots, \Pi_7$ such that $|\Pi_{2m}| = \lceil \frac{k}{2} \rceil$, and $|\Pi_{2m+1}| = \lfloor \frac{k}{2} \rfloor$ for $m \in \{0, \ldots, 3\}$.

These $2n$ processors are now connected similarly as in the proof of Lemma 7:

The pairwise communication channels are reconnected such that each processor of block Π_i sends his outgoing messages to the processors of the blocks Π_{i-1}, Π_{i+1}, and Π_{i+2}, while interpreting the indices modulo 8 instead of modulo 4.

The two-cast channels are reconnected such that each processor of block Π_i two-casts his outgoing messages to the processor pairs among $\Pi_{i-2} \cup \Pi_{i-1}$, $\Pi_{i-1} \cup \Pi_{i+1}$, and $\Pi_{i+1} \cup \Pi_{i+2}$ while interpreting the indices modulo 8.

The rest of the proof proceeds analogously to the proof of Lemma 7 by arguing about the consistency among adjacent blocks Π_i of processors rather than only among adjacent (single) processors π_i. ■

4 Equivalence of consistency primitives

Theorem 4 states that two-cast implies broadcast for any $n \geq 3$ and $t < n/2$. This result can be generalized by proving equivalence of a large class of consistency primitives, i.e., that any single primitive from this class can be used to efficiently simulate any other one from this class. First it is shown that even weak broadcast (the dealer variant of weak consensus (Definition 3)) among three players, called *weak two-cast*, is sufficient in order to achieve broadcast for n and $t < n/2$. Second we prove that, more generally, any broadcast (or weak broadcast) primitive for n_0 players that is resilient against $t_0 = \lceil \frac{n_0}{3} \rceil$ player corruptions is sufficient. Finally, these results are extended to consensus, yielding the following theorem whose proof immediately follows from Theorem 4 and Lemmas 8, 9, and 10.

Theorem 6 *The following consistency primitives are equivalent (up to a simulation cost polynomial in the number of players n):*

- *weak broadcast for any $n \geq 3$ with $t = \lceil n/3 \rceil$.*
- *weak consensus for any $n \geq 3$ with $t = \lceil n/3 \rceil$.*
- *two-cast with $t \leq 3$.*
- *broadcast for any n with $t < n/2$.*
- *consensus for any n with $t < n/2$.*

4.1 Weak broadcast

In weak broadcast, the dealer holds an input $v \in \{0,1\}$ and every player decides on a value $v' \in \{0,1,2\}$.

Definition 4: A protocol achieves *weak broadcast* if it satisfies the following conditions.

[6] Note that, a priori, we assume that any input value from $\{0,1\}$ will be selected with some non-negligible probability σ by the dealer. Otherwise the broadcast problem could be trivially solved for any $t \leq n$ by a protocol wherein every player decides on the value that is selected with overwhelming probability.

Consistency: If $v'_p \in \{0,1\}$ for any correct player p then $v'_q \in \{v'_p, 2\}$ for every correct player q.

Validity: If the dealer is correct then every correct player p decides on the dealer's input value: $v'_p = v$.

Lemma 8 *Weak two-cast implies two-cast with a constant simulation cost.*

Proof: Given weak two-cast, two-cast can be implemented as follows: First, the dealer distributes his value by a weak two-cast protocol. Then both receivers exchange the values they have received from the dealer. A receiver who received a value $v < 2$ from the dealer sticks to this value whereas in the other case ($v = 2$) he replaces his value by the value w received from the other receiver (during the second round) or on 0 if $w = 2$.
Hence, if the dealer is correct, a correct receiver always decides on the dealer's value. On the other hand, if the dealer is corrupted, then two correct receivers either receive the same value $v \in \{0,1\}$ or at least one of them receives $v = 2$ (by the consistency property of weak two-cast) which makes him adapt the other players' value w if $w \in \{0,1\}$. Finally, if the weak broadcast results in $v = 2$ for both receivers then both of them replace their values by 0. ∎

4.2 Even weaker broadcast

For our constructions, we always assumed two-cast or weak two-cast to be reliable independently of the number of corrupted players that are involved. In fact, the same constructions even work with two-cast or weak two-cast that is only secure against one player corruption (whereas nothing is assumed about an invocation of this primitive if more than one player is corrupted). More generally, we show that any broadcast or weak broadcast primitive for n_0 players that is resilient against $t_0 = \lceil \frac{n_0}{3} \rceil$ player corruptions is sufficient in order to achieve two-cast among $n = 3$ players with $t \leq 3$, and hence to achieve broadcast for any $n \geq 3$ with $t < n/2$.

Lemma 9 *Weak broadcast for any $n_0 \geq 3$ with $t_0 = \lceil n_0/3 \rceil$ implies weak two-cast.*

Proof: The proof proceeds in two steps. Weak broadcast for any $n_0 \geq 3$ with $t_0 = \lceil n_0/3 \rceil$ is first reduced to weak two-cast that tolerates $t \leq 1$ player corruptions. In a second step, this two-cast primitive is generalized to tolerate arbitrarily many player corruptions.

1. In order to achieve weak two-cast among three players p_1, p_2, and p_3, we let each of these players simulate (any) up to $\lceil n_0/3 \rceil$ players in the given weak broadcast protocol (with the only restriction that the dealer of the weak two-cast in fact simulates the dealer of the weak broadcast). Since, by assumption, at most one of the players p_i ($i \in \{1,2,3\}$) is corrupted who simulates at most $\lceil n_0/3 \rceil$ players in the original protocol, the original protocol achieves broadcast among the simulated players. Hence we can let each player p_i decide on the value of any one of his simulated players.
2. Any weak two-cast with $t \leq 1$ can be extended to tolerate arbitrarily many player corruptions. Since the communication model is synchronous there is an upper bound on the delay time on every underlying communication primitive. Hence there is an upper bound on the delay time of the given (possibly composed) two-cast whenever only $t \leq 1$ players are actually corrupted. Hence, in order to tolerate $t \leq 3$, we can let the players invoke the same two-cast primitive with the only restriction that any receiver sticks to some default value as soon as the upper bound on the delay time is exceeded. Since, in this case, at most one player is correct, all conditions for weak broadcast are trivially satisfied.

∎

4.3 Consensus primitives

Independently of the model, the achievability of consensus always implies achievability of broadcast since, given a consensus protocol, we can let the dealer multicast his input value in a first phase and then let all players run the consensus protocol on the received values. On the other hand, the achievability of broadcast implies the achievability of consensus whenever the corrupted players form a minority, since we can use broadcast for every player to publish his input value in a first phase, and in a second phase, the players perform a majority voting on all received values. The same argumentation holds for the mutual implication of weak broadcast and weak consensus, and hence we get the following lemma:

Lemma 10 *Given n players and $t < n/2$ then broadcast and consensus are equivalent, and weak broadcast and weak consensus are equivalent (up to a simulation cost polynomial in n).*

5 Secure multi-party computation

As a more general task than broadcast or consensus, secure multi-party computation allows the players to distributedly compute an arbitrary function on the player's inputs by keeping the player's inputs private and guaranteeing correctness of the computation. Ben-Or, Goldwasser, and Wigderson [2], and Chaum, Crépeau, and Damgård [4] proved that in the standard model with a synchronous network of pairwise authentic channels unconditionally secure multi-party computation among n players is possible if and only if $t < n/3$ of the players are actively corrupted. Rabin and Ben-Or [18] later proved that when additionally assuming global broadcast channels, unconditionally secure multi-party is even achievable if (and only if) $t < n/2$.
Our results now immediately imply that the same bound is achievable under the considerably weaker assumption of weak broadcast for only three players each, which is stated in the following theorem.

Theorem 7 *Given weak broadcast among each triple of players, unconditionally secure multi-party computation among n players is possible if and only if the number t of actively corrupted players satisfies $t < n/2$. There exist protocols with communication and computation complexities polynomial in n.*

Proof: ($\Longleftarrow$): In order to achieve multi-party computation among n players secure against $t < n/2$ active player corruptions, the protocol of either [18] or [6] is applied by additionally substituting every invocation of the global broadcast channel by the broadcast protocol that was constructed in Section 2, and by then simulating two-cast by weak two-cast according to the proof of Lemma 8. The efficiency of this protocol immediately follows from the efficiency of the protocols in [18, 6] and from the efficiency of the constructions in Section 2 and in the proof of Lemma 8.

($\Longrightarrow$): If secure multi-party computation would be achievable for any $t \geq n/2$ then especially broadcast would be achievable as a special case of general multi-party computation, in contradiction to Theorem 5. ∎

6 Conclusion and open problems

We have shown that, when assuming certain weak consistency primitives in addition to the standard communication model, broadcast and consensus among n players is achievable whenever $t < n/2$ instead of $t < n/3$ in the standard model. Moreover, a large class of such consistency primitives is equivalent. For a further line of research it would be interesting to find achievability reductions including additional consistency primitives. For example, one concrete open question is to characterize what is achievable when extending the standard model with broadcast among $n_0 > 3$ players that tolerates any number of player corruptions (instead of $n_0 = 3$ in the two-cast model). Furthermore it would be interesting to know whether the same results still hold with respect to an incomplete two-cast network where only a subset of all two-cast channels is assumed, i.e., to characterize tight conditions on the network of two-cast channels for the previous results still being achievable.

Acknowledgments

The authors would like to thank Ronald Cramer, Faith Fich, Juan Garay, Nicolas Gisin, and Martin Hirt for many interesting discussions, and Juan Garay for suggesting the term two-cast for broadcast among three players.

References

[1] Amotz Bar-Noy, Danny Dolev, Cynthia Dwork, and H. Raymond Strong. Shifting gears: Changing algorithms on the fly to expedite Byzantine agreement. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, pages 42–51, Vancouver, British Columbia, Canada, 10–12 August 1987.

[2] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 1–10, 1988.

[3] Piotr Berman, Juan A. Garay, and Kenneth J. Perry. Towards optimal distributed consensus (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 410–415, Research Triangle Park, North Carolina, 30 October–1 November 1989. IEEE.

[4] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 11–19, 1988.

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

[6] Ronald Cramer, Ivan Damgård, Stefan Dziembowski, Martin Hirt, and Tal Rabin. Efficient multiparty computations secure against an adaptive adversary. In *Advances in Cryptology — EUROCRYPT '99*, Lecture Notes in Computer Science, 1999.

[7] Danny Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, 1982.

[8] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *Journal of the ACM*, 40(1):17–47, January 1993.

[9] Danny Dolev, Michael J. Fischer, Rob Fowler, Nancy A. Lynch, and H. Raymond Strong. An efficient algorithm for Byzantine agreement without authentication. *Information and Control*, 52(3):257–274, March 1982.

[10] Pesech Feldman and Silvio Micali. Optimal algorithms for Byzantine agreement. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pages 148–161, 1988.

[11] Matthew Franklin and Moti Yung. Secure hypergraphs: Privacy from partial broadcast (extended abstract). In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 36–44, Las Vegas, Nevada, 29 May–1 June 1995.

[12] Matthew K. Franklin and Rebecca N. Wright. Secure communication in minimal connectivity models. In Kaisa Nyberg, editor, *Advances in Cryptology: EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*. Springer, 1998.

[13] Juan A. Garay and Yoram Moses. Fully polynomial Byzantine agreement in $t + 1$ rounds (extended abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 31–41, San Diego, California, 16–18 May 1993.

[14] Juan A. Garay and Kenneth J. Perry. A continuum of failure models for distributed computing. In Adrian Segall and Shmuel Zaks, editors, *Distributed Algorithms, 6th International Workshop, WDAG '92*, volume 647 of *Lecture Notes in Computer Science*, pages 153–165, Haifa, Israel, 2–4 November 1992. Springer.

[15] Anna Karlin and Andrew C. Yao. Manuscript.

[16] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann series in data management systems. Morgan Kaufmann Publishers, Los Altos, CA 94022, USA, 1996.

[17] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, April 1980.

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

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

[20] Russell Turpin and Brian A. Coan. Extending binary Byzantine Agreement to multivalued Byzantine Agreement. *Information Processing Letters*, 18(2):73–76, February 1984.