# Byzantine Agreement Secure Against General Adversaries in the Dual Failure Model*

Bernd Altmann and Matthias Fitzi and Ueli Maurer
{altmann,fitzi,maurer}@inf.ethz.ch

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

**Abstract.** This paper introduces a new adversary model for Byzantine agreement and broadcast among a set $P$ of players in which the adversary may perform two different types of player corruption: active (Byzantine) corruption and fail-corruption (crash). As a strict generalization of the results of Garay and Perry, who proved tight bounds on the maximal number of actively and fail-corrupted players, the adversary's capability is characterized by a set $\mathcal{Z}$ of pairs $(A, F)$ of subsets of $P$ where the adversary may select an arbitrary such pair $(A_i, F_i)$ from $\mathcal{Z}$ and corrupt the players in $A_i$ actively and fail-corrupt the players in $F_i$.
For this model we prove that the exact condition on $\mathcal{Z}$ for which perfectly secure agreement and broadcast are achievable is that for no three pairs $(A_i, F_i)$, $(A_j, F_j)$, and $(A_k, F_k)$ in $\mathcal{Z}$ we have $A_i \cup A_j \cup A_k \cup (F_i \cap F_j \cap F_k) = P$. Achievability is demonstrated by efficient protocols. Moreover, for a slightly stronger condition on $\mathcal{Z}$, which covers the previous mixed (active and fail-corruption) threshold condition and the previous purely-active non-threshold condition, we demonstrate agreement and broadcast protocols that are substantially more efficient than all previous protocols for these two settings.

**Key words.** Broadcast, Byzantine agreement, unconditional security, active adversary, fail-corruption.

## 1 Introduction

Byzantine agreement and broadcast are two closely related fundamental problems in distributed systems and cryptography, in particular in secure multi-party computation. In this paper we consider Byzantine agreement (and broadcast) protocols among a set of players in a standard model with a complete synchronous network of pairwise authenticated channels among the players.

### 1.1 Player Corruption

We demand the protocols to be perfectly secure (i.e. unconditionally secure with no probability of error) against an adversary that may corrupt players in two different ways:

* Published in *Distributed Computing — DISC '99*, Lecture Notes in Computer Science, vol. 1693, Springer. Research supported by the Swiss National Science Foundation (SNF), SPP project no. 5003-045293.

**Active corruption:** The adversary takes full control over the corrupted players and makes them deviate from the protocol in an arbitrary way.
**Fail-corruption:** At an arbitrary time during the protocol, chosen by the adversary, the communication from and to the corrupted player is stopped.

The players that are fail-corrupted or uncorrupted are called *non-malicious* since they do not deviate from the protocol as long as they participate. A player is called *correct* if he is non-malicious and has not failed yet at the described point of time. Thus correctness describes a temporal property of the players and only a player that is correct at the end of the protocol is actually uncorrupted.

For a fail-corrupted player to *fail during some communication round* means that he is correct up to this point and, during this round, stops communicating with at least one correct player.[1] The player sends no messages during any subsequent round of the protocol.

### 1.2 Byzantine Agreement and Broadcast

A *Byzantine agreement* protocol is defined for a set of $n$ players with every player initially holding an input value and finally deciding on an output value such that the following conditions are satisfied:

**Agreement:** All uncorrupted players decide on the same output value.
**Validity:** If all initially correct players hold the same input value $v$ then all uncorrupted players decide on $v$.
**Termination:** For all non-malicious players the protocol terminates after a finite number of rounds.

In contrast to agreement, *broadcast* is defined with respect to one particular player called the dealer who initially inputs a value. Again, every player decides on some output value. For broadcast the former agreement and termination conditions are still required. The validity condition transforms into

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

Note that it suffices to focus on bit-agreement (or bit-broadcast) protocols where the domain of values is restricted to $\{0,1\}$ since protocols for any finite domain with cardinality $m$ can be easily obtained by applying $\lceil \log m \rceil$ bit-protocols in parallel. This does not change the round complexity and increases the communication complexity only by a factor $\lceil \log m \rceil$.

### 1.3 Previous Work

In the threshold model with an active adversary, Lamport, Shostak and Pease [PSL80,LSP82] proved that Byzantine agreement is achievable if and only

[1] Our model includes that in the round during which a player fails, some correct players may still receive a valid message by this player whereas others may not. Hence the correct players' views about which players have failed can be inconsistent, and this must be taken into account in the design and analysis of the protocols.

if less than one third of the players are actively corrupted ($t < n/3$). For this model numerous protocols with optimal resilience have been proposed in the literature [DFF$^+$82,BDDS87,TPS87,FM88,BGP89,CW92,GM93], which all have communication and computation complexities polynomial in the number $n$ of players. In the threshold model with an adversary that may only perform fail-corruptions, Lamport and Fischer [LF82] proved that agreement is achievable for any $t < n$.

These results have been unified in [GP92] for the threshold model where an adversary is considered who may corrupt arbitrary $t$ players, but at most $b \leq t$ of them actively (the rest is only fail-corrupted). They proved that $t+2b < n$ is a tight bound on agreement to be achievable and proposed protocols with optimal resilience and polynomial complexities.

In the more general context of secure multi-party computation, Hirt and Maurer [HM97] introduced the concept of a general adversary that is characterized by an adversary structure which is a set of subsets of the player set. The adversary may corrupt the players of exactly one of these subsets. For the same model, with respect to an active adversary, Fitzi and Maurer [FM98] proposed optimally resilient broadcast protocols with computation complexity polynomial in the size of the adversary structure and communication complexity polynomial in the number $n$ of players.

### 1.4 Contributions

This paper unifies the models of [GP92] and [FM98] to the new model with a general adversary that may simultaneously corrupt some players actively and some other players to fail. For this model a tight condition on the adversary structure is proven for Byzantine agreement to be achievable. Efficient protocols are proposed for every structure that meets this condition. This condition for example guarantees that, quite surprisingly, agreement is possible among four players $p_1$, $p_2$, $p_3$, and $p_4$ if any player $p_i$ is actively corrupted and all the remaining players except $p_{((i+1) \bmod 4)}$ are fail-corrupted.

Furthermore we present a protocol that, when restricting this model to the special cases of [GP92] and [FM98], is even more efficient than any protocol previously known for these special cases.

Although all these results (tight condition and protocols) are only presented for agreement they immediately hold for broadcast as well since, with only minor modifications, a broadcast protocol can be easily obtained from any agreement protocol and vice versa. Our proposed agreement protocols can even be turned into a broadcast protocol with no loss of efficiency.

### 1.5 Definitions and Notation

The *player set* is denoted by $P = \{p_1, \ldots, p_n\}$ and its *cardinality* by $n = |P|$. The adversary is defined by a *general adversary structure* $\mathcal{Z}$ which is a set of *classes* $(A, F)$ with $A \subseteq P$ and $F \subseteq P$ where the players of exactly one class $(A, F)$ may be corrupted — actively corrupted for the players in $A$ and corrupted to fail for the players in $F$. Without loss of generality we demand $A \cap F = \emptyset$ since active corruption is strictly more general than fail-corruption.

A class $(A', F')$ is *contained* in a class $(A, F)$ (written $(A', F') \subseteq (A, F)$) if $A' \subseteq A$ and $F' \subseteq A \cup F$, and a class $(A', F')$ is *strictly contained* in a class $(A, F)$ if it is contained, and $A' \subset A$ or $F' \subset F$ (written $(A', F') \subset (A, F)$).

The adversary structure $\mathcal{Z}$ is defined to be *monotone* with respect to inclusion, i.e.,

$$(A, F) \in \mathcal{Z} \wedge (A', F') \subseteq (A, F) \Longrightarrow (A', F') \in \mathcal{Z}.$$

The *basis* $\overline{\mathcal{Z}}$ of an adversary structure $\mathcal{Z}$ is the set of all maximal elements of $\mathcal{Z}$

$$\overline{\mathcal{Z}} = \left\{ (A, F) \in \mathcal{Z} \mid \nexists (A', F') \in \mathcal{Z} : (A, F) \subset (A', F') \right\}.$$

A set $A$ is called an *active set* of an adversary structure $\mathcal{Z}$ if $(A, \emptyset) \in \mathcal{Z}$. A set $F$ is called a *fail set* of $\mathcal{Z}$ if $(\emptyset, F) \in \mathcal{Z}$. The set of all active sets of an adversary structure $\mathcal{Z}$ is denoted by $\mathcal{Z}_A$.

The following predicates $Q(P, \mathcal{Z})$ and $R(P, \mathcal{Z})$ on adversary structures $\mathcal{Z}$ with respect to a player set $P$ will be needed later in this paper:

$$Q(P, \mathcal{Z}) := \forall (A_1, F_1), (A_2, F_2), (A_3, F_3) \in \mathcal{Z} : A_1 \cup A_2 \cup A_3 \cup F_1 \neq P.$$

$$R(P, \mathcal{Z}) := \forall (A_1, F_1), (A_2, F_2), (A_3, F_3) \in \mathcal{Z} : A_1 \cup A_2 \cup A_3 \cup (F_1 \cap F_2 \cap F_3) \neq P.$$

Note that $Q(P, \mathcal{Z})$ implies $R(P, \mathcal{Z})$ and that, because of symmetry, $Q(P, \mathcal{Z})$ is equivalent to $R(P, \mathcal{Z})$ in the threshold case.

## 2 Necessary Condition

Given the tight bound of $t+2b < n$ for the threshold model[2] in [GP92] it might be obvious to conclude that $Q(P, \mathcal{Z})$ is a necessary condition for the general model, i.e., for any three classes in $\mathcal{Z}$ the union of all active sets with one of the fail sets must not cover the full player set $P$. However, as a consequence of the (generally) asymmetric properties of general adversary structures, agreement can still be achievable when this bound is violated since the failure of one particular player may rule out certain classes of the structure to be selected by the adversary. Thus only a weaker condition can be proven to be necessary which we prove to be tight in the next section: $R(P, \mathcal{Z})$, i.e., for any three classes in $\mathcal{Z}$ the union of all active sets with the intersection of all fail sets must not cover the full player set $P$.

**Theorem 1.** *For a set $P$ of $n$ players and any adversary structure $\mathcal{Z}$, $R(P, \mathcal{Z})$ is necessary for Byzantine agreement to be achievable.*

*Proof.* For the sake of contradiction suppose that there is an agreement protocol for some adversary structure $\mathcal{Z}$ violating $R(P, \mathcal{Z})$. Hence there are three classes $(A_1, F_1), (A_2, F_2), (A_3, F_3) \in \mathcal{Z}$ such that $A_1 \cup A_2 \cup A_3 \cup (F_1 \cap F_2 \cap F_3) = P$. Due to the monotonicity of the adversary structure we can assume without loss of generality that $F_1 = F_2 = F_3 =: F$ and that the sets $A_1$, $A_2$, $A_3$ and $F$ are pairwise disjoint. One possible strategy for the adversary is to make all players

[2] i.e. three times the number of actively corrupted plus once the number of fail-corrupted players must be less than $n$

in $F$ fail at the beginning of the protocol. Hence this protocol can be easily modified into a secure agreement protocol for the player set $P' = P \setminus F$ with respect to $\mathcal{Z}$ restricted to the players in $P'$ since by assumption this protocol is correct even if no player in $F$ ever sends any message. Since $A_1 \cup A_2 \cup A_3 = P'$ this contradicts the result of [PSL80,HM97] that agreement is impossible in this case. □

## 3 Optimally Resilient Protocols

This section describes protocols for any player set $P$ and adversary structure $\mathcal{Z}$ satisfying $R(P, \mathcal{Z})$.

### 3.1 Protocol Elements and Code Notation

The protocols are constructed basically along the lines of the protocols of [GP92] which are based on several subprotocols. The main idea of these subprotocols is that every player enters them with his *preferred* value he inclines to decide on and exits them with an updated (potentially different) preferred value such that the following two conditions are satisfied

**Persistence:** If all correct players enter the subprotocol with the same preferred value then, after the execution of the subprotocol, all correct players[3] still prefer this value. In other words, the subprotocol has no effect when agreement had previously been achieved.

**Consistence:** In any case (also if the correct players enter the subprotocol with distinct preferred values) the values preferred by the correct players at the end of the subprotocol are consistent (in a way to be defined separately for each particular subprotocol).

The effect of such subprotocols can be interpreted as getting the correct players closer to a state of agreement whereas, once achieved, agreement cannot be reversed anymore by the corrupted players. In this paper, often a weaker form of consistence is used that depends on whether some fail-corrupted (i.e. non-actively corrupted) player fails during the execution of the according subprotocol.

**Conditional consistence:** Consistence provided that no fail-corrupted player fails during the execution of the subprotocol in consideration.

All pseudo code descriptions of protocols are stated with respect to the local view of one particular player. 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 $p$. For every player $p$, two global[4] variables are used throughout all subprotocols, $v_p$ and $L_p$. $v_p$ denotes the preferred value by player $p$. $L_p$ is a set in which player $p$ collects all players that he has detected to be corrupted (active or fail). $L$ is initialized to the empty set and (for a correct player) will never contain any correct player.

[3] i.e. all players who are *still* correct — remember the temporal definition of correctness.

[4] with respect to $p$'s scope

### 3.2 Value Unification

This section describes the crucial subprotocol of the agreement protocol: `MakeUnique`. It satisfies the persistence property according to Section 3.1 and conditional consistence in a way that at the end no two correct players prefer distinct values in $\{0, 1\}$ if no fail-corrupted player fails during the execution of the subprotocol. In order to achieve this the original bit domain is extended by an invalidity value 2. However, the preferred value $v$ is still required to be in $\{0, 1\}$ *before* the execution of `MakeUnique`.

`MakeUnique`:

1. `SendToAll`$(v)$; // i.e. send $v$ to every other player
2. $L := L \cup \{ r \in P \mid \text{no value received from } r \text{ or value outside } \{0,1\} \}$;
3. $C^0 := \{ r \in P \mid r \text{ sent } 0 \} \setminus L$;
4. $C^1 := \{ r \in P \mid r \text{ sent } 1 \} \setminus L$;
5. if $(C^1, L) \in \mathcal{Z}$ then $v := 0$
6. elseif $(C^0, L) \in \mathcal{Z}$ then $v := 1$
7. else $v := 2$
8. fi;

If, at the end of `MakeUnique`, a player $p$ holds some value $v_p \in \{0, 1\}$ we say that player $p$ accepts value $v_p$. On the other hand $v_p = 2$ means that player $p$ rejects any value from $\{0, 1\}$. More precisely, a correct player accepts a value $v \in \{0, 1\}$ exactly if, according to his view, agreement on $v$ could have been achieved before the execution of `MakeUnique`. For $v \in \{0, 1\}$ we define $\overline{v}$ as $\overline{v} := 1 - v$.

**Lemma 1 (Persistence of `MakeUnique`).** *If all correct players initially hold the same value $v \in \{0, 1\}$, then after the execution of* `MakeUnique` *every correct player $p$ holds the value $v_p = v$.*

*Proof.* Let $p$ be a player who is correct at the end of `MakeUnique`. Since all correct players initially hold the same value $v$ every such player either sends this value to $p$ or fails during this communication round. Hence $\overline{v}$ will only be received from an actively corrupted player, and $(C_p^{\overline{v}}, L_p) \in \mathcal{Z}$ holds. Hence also $(C_p^v, L_p) \notin \mathcal{Z}$ must hold since otherwise $R(P, \mathcal{Z})$ would be violated (because $P = C_p^v \cup C_p^{\overline{v}} \cup (L_p \cap L_p)$). Thus $v_p = v$ after the execution of `MakeUnique`. □

**Lemma 2 (Conditional Consistence of `MakeUnique`).** *If after the execution of* `MakeUnique`*, two correct players $p$ and $q$ hold values $v_p \neq 2$ and $v_q \neq 2$, respectively, then either $v_p = v_q$ or at least one fail-corrupted player failed during the execution of* `MakeUnique`*.*

*Proof.* Suppose for the sake of contradiction that no fail-corrupted player fails between sending his value to the players $p$ and $q$ and that $v_p = v$ and $v_q = \overline{v}$ for some $v \in \{0, 1\}$ and hence $(C_p^{\overline{v}}, L_p) \in \mathcal{Z}$ and $(C_q^v, L_q) \in \mathcal{Z}$. We have $P = C_p^{\overline{v}} \cup C_p^v \cup L_p$ and hence $P$ can be decomposed as

$$C_p^{\overline{v}} \cup \underbrace{(C_p^v \cap C_q^v)}_{\subseteq C_q^v} \cup \underbrace{(C_p^v \setminus C_q^v) \cup (L_p \setminus L_q)}_{=:A} \cup (L_p \cap L_q) = P. \qquad (1)$$

Since no fail-corrupted player failed during the execution of this protocol all players in $A = (C_p^v \setminus C_q^v) \cup (L_p \setminus L_q)$ must be actively corrupted and hence $(A, L_p \cap L_q) \in \mathcal{Z}$ must hold. Thus we have $(C_p^{\overline{v}}, L_p) \in \mathcal{Z}$, $(C_q^v, L_q) \in \mathcal{Z}$ and $(A, L_p \cap L_q) \in \mathcal{Z}$, and by Equation (1) $C_p^{\overline{v}} \cup C_q^v \cup A \cup (L_p \cap L_q) = P$, which contradicts $R(P, \mathcal{Z})$. $\square$

### 3.3 Agreement Protocol

The agreement protocol consists of a loop over a sequence of statements where one single iteration of the loop can be interpreted in the following way:

The players run the `MakeUnique` protocol in order to guarantee that no two correct players continue with distinct values in $\{0,1\}$. In the next round all players report their (unified) values to every other player. Then every player accepts a value $v \in \{0,1\}$ if, according to his view, at least one correct player reported on $v$ — otherwise he rejects by setting $v := 2$. Finally some distinguished player, called the *king*, reports on his particular value $v$ which is adopted exactly by those players who know that at least one correct player rejected after `MakeUnique` (which implies that agreement did not hold before this particular iteration of the loop).

`Agreement(VAR` $v$`)` **(Agreement Protocol 1)**:

$L := \emptyset$;<br>
for $i := 1$ to $\lceil \log n \rceil n$ do<br>
    $k := ((i-1) \ mod \ n) + 1$; // Assign king<br>
    `MakeUnique`; // Communication Phase 1<br>
    `SendToAll(`$v$`)`; // Communication Phase 2<br>
    $L := L \cup \{ r \in P \mid \text{no value received from } r \text{ or value outside } \{0,1,2\} \}$;<br>
    $D^i := \{ r \in P \mid r \text{ sent } i \} \setminus L$ for $i \in \{0,1,2\}$;<br>
    if $(D^0, L) \notin \mathcal{Z}$ then $v := 0$<br>
      elseif $(D^1, L) \notin \mathcal{Z}$ then $v := 1$<br>
      else $v := 2$<br>
    fi;<br>
    $p_k$ (only): `SendToAll(`$v$`)`; // Communication Phase 3<br>
    $w :=$ value received by $p_k$; // if no value is received then set $w := 0$<br>
    if $(D^2, L) \notin \mathcal{Z}$ then $v := \min(1, w)$ fi;<br>
od;

Every single iteration of the for-loop can be seen as a subprotocol with persistence and conditional consistence properties according to Section 3.1. These properties are stated in the next lemmas.

**Lemma 3 (Conditional Consistence).** *At the end of any iteration of the for-loop with a correct king $p_k$ during which no fail-corrupted player fails, every correct player $p$ holds the same value $v_p = v$.*

*Proof.* Consider some $k^{th}$ iteration of the for-loop with $p_k$ being correct and during which no fail-corrupted player fails. If all correct players replace their values $v$ by $\min(1, w)$, we are done since all correct players receive the same value $w$ from player $p_k$.

Suppose now that at least one correct player $p$ ignores the value sent by the king since $(D_p^2, L_p) \in \mathcal{Z}$ holds. Hence $v_p \neq 2$ since otherwise, according to the Lines 8 to 10 of the protocol, also $(D_p^0, L_p) \in \mathcal{Z}$ and $(D_p^1, L_p) \in \mathcal{Z}$ would hold in contradiction to $R(P, \mathcal{Z})$. Let $v := v_p$. $(D_p^v, L_p) \notin \mathcal{Z}$ ($v \neq 2$ and Lines 8 to 10) implies that at least one correct player sent $v$ during Communication Phase 2 and, due to the (conditional) consistence of `MakeUnique`, no correct player sent $\overline{v}$ during the same phase. Hence, for every correct player $q$, $D_q^{\overline{v}}$ can only contain actively corrupted players and hence $(D_q^{\overline{v}}, L_q) \in \mathcal{Z}$ holds.

Suppose, for the sake of contradiction, that there is a correct player $q$ who enters Communication Phase 3 with a value $v_q \neq v$, i.e., $(D_q^v, L_q) \in \mathcal{Z}$. $P = D_p^{\overline{v}} \cup D_p^v \cup D_p^2 \cup L_p$ can be decomposed as

$$\underbrace{D_p^{\overline{v}} \cup (D_p^v \setminus D_q^v)}_{=:A_1} \cup \underbrace{(D_p^v \cap D_q^v)}_{\subseteq D_q^v} \cup D_p^2 \cup \underbrace{(L_p \setminus L_q)}_{=:A_2} \cup (L_p \cap L_q) = P.$$

All players in $A = A_1 \cup A_2$ are actively corrupted since they sent either $\overline{v}$ or distinct values to the players $p$ and $q$ or failed[5] in $p$'s view but not in $q$'s view. Hence $(A, L_p \cap L_q) \in \mathcal{Z}$ which leads to a contradiction with condition $R(P, \mathcal{Z})$ since together with $(D_p^2, L_p) \in \mathcal{Z}$ and $(D_q^v, L_q) \in \mathcal{Z}$ we have $D_p^2 \cup D_q^v \cup A \cup (L_p \cap L_q) = P$.

Thus every correct player $q$ enters Communication Phase 3 with $(D_q^v, L_q) \notin \mathcal{Z}$ and hence $v_q = v = v_p$. Since especially the king $p_k$ is correct every player who accepts $p_k$'s value accepts $v_{p_k} = v = v_p$. □

**Lemma 4 (Persistence).** *If at the beginning of any iteration of the for-loop every correct player $p$ holds the same value $v_p = v \neq 2$, then every correct player holds $v$ at the end of the iteration even if some fail-corrupted players fail.*

*Proof.* Due to the persistence property of `MakeUnique` (Lemma 1) every correct player $p$ holds $v_p = v$ after `MakeUnique` and hence, after `SendToAll`, $(D_q^{\overline{v}}, L_q) \in \mathcal{Z}$ and $(D_q^2, L_q) \in \mathcal{Z}$. Because of the condition $R(P, \mathcal{Z})$ also $(D_q^v, L_q) \notin \mathcal{Z}$ must hold. Thus every correct player $p$ ignores the king in Communication Phase 3 and holds value $v_p = v$ at the end of the loop. □

The following theorem together with Theorem 1 shows that the condition $R(P, \mathcal{Z})$ is tight:

**Theorem 2.** *For a set $P$ of $n$ players and an adversary structure $\mathcal{Z}$ perfectly secure Byzantine agreement is achievable if $R(P, \mathcal{Z})$ is satisfied. For every structure $\mathcal{Z}$ satisfying $R(P, \mathcal{Z})$ there is such a protocol with communication complexity polynomial in $n$ and computation complexity polynomial in $|\overline{\mathcal{Z}}|$.*[6]

*Proof.* We first show by contradiction that, in Agreement Protocol 1, all uncorrupted players finally decide on the same value. Thus assume that two uncorrupted players decide on distinct values. Then, according to Lemma 3, there was

[5] note that we suppose no fail-corrupted player to fail during this loop

[6] Under the natural assumption that there exists an algorithm polynomial in $n$ to decide whether a given class $(A, F)$ is an element of the adversary structure $\mathcal{Z}$, the computation complexity is also polynomial in $n$.

no iteration of the for-loop with a correct king during which no fail-corrupted player failed. Let $C(0) = P$ and $C(i)$ denote the set of players that are still correct at the end of iteration $i$ of the for-loop, and let $c(0) = n$ and $c(i) = |C(i)|$. We argue that during any $n$ sequential iterations $i = j, \ldots, j+n-1$ at least $c(j-1)/2$ fail-corrupted players failed. The failure of one single fail-corrupted player can prevent agreement for at most two iterations with a king from the set $C(j-1)$ — one correct king's iteration and his own one[7]. Hence at least half of the players in $C(j-1)$ must have failed. Hence, for any $l$ with $0 < l \leq \lceil \log n \rceil$, $c(ln) \leq c((l-1)n)/2$ and $c(ln) \leq c(0)/2^l$, i.e. for $l = \lceil \log n \rceil$ (after the last iteration of the for-loop) we have

$$c(\lceil \log n \rceil n) \leq c(0)/2^{\lceil \log n \rceil} \leq c(0)/n = 1,$$

in contradiction to the fact that at least two players are uncorrupted and hence $c(\lceil \log n \rceil n) > 1$. Hence there is a first iteration of the for-loop with a correct king during which no fail-corrupted player fails. After this iteration agreement holds (Lemma 3) and due to Lemma 4 agreement holds also at the end of the protocol.

The validity and termination properties are obviously satisfied. The efficiency can be easily verified by code inspection. □

## 4 Efficiency Improvements by Early Stopping

A major disadvantage of Agreement Protocol 1 of Section 3 is that the players must continue to iterate the for-loop even if agreement on some value has already been reached. The goal of this section is to derive a protocol that can be terminated as soon as agreement is achieved, i.e., a protocol that terminates early if only few players are corrupted. This is achieved by some modifications of Agreement Protocol 1.

However, a full description and correctness proof of our early stopping protocols for condition $R(P, \mathcal{Z})$ would exceed the limits of this extended abstract. Instead, we give an early stopping protocol with respect to the stronger condition $Q(P, \mathcal{Z})$ which can be handled more easily.[8] Moreover these protocols, when applied to the case of a mixed (active and fail-corruption) threshold adversary [GP92] or a purely-active non-threshold adversary [FM98], are even more efficient than any previously known protocols for these special cases.[9]

### 4.1 Protocol Modifications

As a consequence of the somewhat stronger condition $Q(P, \mathcal{Z})$ on $P$ and $\mathcal{Z}$, explicit failure detection becomes unnecessary (i.e. $L$ drops out of the algorithms). The main idea is to achieve the following property which is important for the correctness of the protocol:

[7] after he has already failed during the correct player's iteration

[8] Note that $Q(P, \mathcal{Z})$ still implies the achievability bounds for [GP92,FM98].

[9] in contrast to our early stopping protocols for $R(P, \mathcal{Z})$ (not described in this extended abstract), which are less efficient than those of [GP92] for their special model.

**Stop-Implication**: A correct player (only) stops early if it is guaranteed that every correct player already prefers the same value $v$ and if it is guaranteed that even after his early stopping $v$ is preferred by every correct player.

Since the Stop-Implication is a property of the final agreement protocol its correctness is proven only later, in the proof of Lemma 10. In order for the subprotocols to still satisfy the persistence property, even if correct players stop early, the following rule is introduced.

**Substitution-Rule**: Whenever, during any communication round, a player $p$ expects a value $x$ to be sent by a player $q$ but does not receive any value, then $x$ is set to the value $x_p$ that has been sent by himself during the same communication round.[10]

This rule together with the Stop-Implication guarantees that, after a correct player $p$ has stopped early, every correct player $q$ replaces any future message by $p$ correctly as if $p$ would still participate in the protocol.[11]

### 4.2 Value Unification

Subprotocol `MakeUnique` of Section 3.2 can be simplified when condition $Q(P, \mathcal{Z})$ is satisfied. Since `MakeUnique` will be applied in two different contexts, we use the variable parameter $x$ in the following pseudo-code description.

`MakeUnique(VAR` $x$`)`:

1. `SendToAll(`$x$`)`;
2. $C^0 := \left\{ r \in P \,\middle|\, r \text{ sent } 0 \right\}$;
3. $C^1 := \left\{ r \in P \,\middle|\, r \text{ sent } 1 \right\}$;
4. if $C^1 \in \mathcal{Z}_A$ then $x := 0$
5. elseif $C^0 \in \mathcal{Z}_A$ then $x := 1$
6. else $x := 2$
7. fi;

It is easy to see that together with the Substitution Rule, persistence (according to Lemma 1) is still satisfied. In contrast to the conditional consistence in Lemma 2 even *unconditional* consistence can be proven.

**Lemma 5 (Consistence of `MakeUnique`).** *If, after the execution of* `MakeUnique`*, two correct players $p$ and $q$ hold values $v_p \neq 2$ and $v_q \neq 2$, then $v_p = v_q$.*

*Proof.* Let $p$ and $q$ be two correct players and, for the sake of contradiction, suppose that $v_p = v \neq 2$ and $v_q = 1 - v = \overline{v}$.

[10] This substitution value is well-defined since communication is symmetric in the sense that during any specific round all players report on their particular view of the same variable or fact. The only exception is the king's round wherein only the king sends his preferred value. In this case simply the own preferred value is taken.

[11] Whenever it is argued that every correct player behaves in a certain way, only players that have not stopped yet are considered.

Suppose that no correct player has stopped so far during the protocol and let $A$ (and $F$) be the sets of players that are actively corrupted (and fail corrupted), and hence $C_p^{\overline{v}} \in \mathcal{Z}_A$, $C_q^v \in \mathcal{Z}_A$ and $(A, F) \in \mathcal{Z}$. Since a correct player sends the same value (in $\{0,1\}$) to both players $p$ and $q$ we have $C_p^{\overline{v}} \cup C_q^v \cup A \cup F = P$ in contradiction to $Q(P, \mathcal{Z})$.

On the other hand, if any correct player has stopped the protocol before then, due to the Stop-Implication, the players $p$ and $q$ hold the same value $v_p = v_q$ after `MakeUnique` because of the persistence property. □

### 4.3 Unicast

In order to enable a player to detect that all correct players prefer the same value (and even will after he stops), Communication Phase 2 of Agreement Protocol 1 is replaced by the more powerful primitive `Unicast`. Note that the for-loop can be parallelized into one communication round.

`Unicast(VAR` $D^0$`,`$D^1$`,`$D^2$`):`

`SendToAll(`$v$`);`
for $l := 1$ to $n$ do
  $R^l :=$ value received from $p_l$;
  $S^l := \begin{cases} 0, \text{ if } R^l \in \{0,1\} \\ 1, \text{ if } R^l = 2 \end{cases}$
  `MakeUnique(`$S^l$`);`
od;
$D^0 := \left\{ p_l \in P \mid R^l = 0 \wedge S^l = 0 \right\}$;
$D^1 := \left\{ p_l \in P \mid R^l = 1 \wedge S^l = 0 \right\}$;
$D^2 := \left\{ p_l \in P \mid R^l = 2 \wedge S^l = 1 \right\}$;
if $D^0 \notin \mathcal{Z}_A$ then $v := 0$
  elseif $D^1 \notin \mathcal{Z}_A$ then $v := 1$
  else $v := 2$
fi;

We say that player $p$ accepts value $v \in \{0,1,2\}$ from player $q$ or, as a short hand, that $p$ accepts $\langle v, q \rangle$ if $q \in D_p^v$. The following lemma follows directly from the consistence and persistence properties of `MakeUnique`.

**Lemma 6 (Consistence of Unicast).** *The value $v$ sent by a correct player $p$ is accepted by every correct player $q$, i.e., $p \in D_q^v$. Moreover, if a correct player $p$ accepts $\langle v, r \rangle$ for $v \in \{0,1\}$ and $r \in P$, then no correct player $q$ accepts $\langle 2, r \rangle$, i.e., $D_q^2 \subseteq (P \setminus D_p^v)$ for any two correct players $p$ and $q$.*

**Lemma 7 (Persistence of Unicast).** *If all correct players prefer the same value $v \in \{0,1\}$ before the execution of* `Unicast` *then all players from which a correct player $p$ does* **not** *accept $v$ are actively corrupted. In particular $(P \setminus D_p^v) \in \mathcal{Z}_A$ (and hence $D_p^2 \in \mathcal{Z}_A$), and $v_p = v$ at the end of* `Unicast`.

*Proof.* According to Lemma 6 the value of every correct player is accepted. It remains to show that even a value is accepted from every fail-corrupted player $p_i$: A correct player either does not receive a value from $p_i$ during `SendToAll` and hence replaces this value by $v$ or he still receives a value from $p_i$ but then this value must be $v$ since $p_i$ was correct until the execution of `Unicast` and therefore preferred $v$ by assumption. Hence $S^i \equiv 0$ for all correct players which persists after `MakeUnique` according to the persistence property. □

### 4.4 Agreement Protocol

`Agreement(VAR` $v$`)` **(Agreement Protocol 2)**:

1. for $k := 1$ to $n$ do
2.     `MakeUnique(`$v$`);` // Communication Phase 1
3.     `Unicast(`$D^0$`,`$D^1$`,`$D^2$`);` // Communication Phase 2
4.     $p_k$ (only): `SendToAll(`$v$`);` // Communication Phase 3
5.     $w :=$ value received from $p_k$;
6.     if $\left(v = 2 \ \vee \ D^2 \notin \mathcal{Z}_A\right)$ then $v := \min(1, w)$
7.       elseif $\left(v \neq 2 \ \wedge \ P \setminus D^v \in \mathcal{Z}_A\right)$ then `stop`
8.     fi;
9. od;

Persistence and consistence can be proven in a similar way as for Agreement Protocol 1 of Section 3. Moreover even *unconditional consistence* can be proven.

**Lemma 8 (Persistence).** *If, at the beginning of some for-loop, all correct players prefer the same value $v \neq 2$ then they still do so at the end of the loop.*

*Proof.* According to the persistence of `MakeUnique` and `Unicast` $D_p^v \notin \mathcal{Z}_A$, $D_p^{\overline{v}} \in \mathcal{Z}_A$ and $D_p^2 \in \mathcal{Z}_A$ are satisfied for every correct player $p$ and hence $v_p = v$ and the king's value is ignored by $p$. □

The following two lemmas (Lemma 9 and 10) are needed for the proof of the consistence property of Agreement Protocol 2. Lemma 10 assures that the Stop-Implication indeed holds — a fact which also the previous proofs for `MakeUnique` and `Unicast` rely on.

**Lemma 9.** *If a correct player $p$ ignores the king's value according to Line 6 of the protocol, then every correct player prefers the same value $v_p$ before Communication Phase 3.*

*Proof.* Suppose that $p$ ignores the king since $v_p = v \neq 2$ (and hence $D_p^v \notin \mathcal{Z}_A$) and $D_p^2 \in \mathcal{Z}_A$ hold. $D_p^v \notin \mathcal{Z}_A$ implies that at least one correct player entered `Unicast` with value $v$ and hence every correct player entered `Unicast` with value $v$ or 2.

For the sake of contradiction suppose that some correct player $q$ enters this phase with some value $v_q \neq v$. Due to Lemma 6 all values by correct players are accepted. Hence, for some set $A$ of actively corrupted players and some set $F$ of fail-corrupted players, we can write the player set as $P = D_p^{\overline{v}} \cup D_p^v \cup D_p^2 \cup A \cup F$ which can be decomposed as

$$\underbrace{D_p^{\overline{v}}}_{=A_1} \cup \underbrace{(D_p^v \setminus D_q^v)}_{=A_2 \cup F_2} \cup \underbrace{(D_p^v \cap D_q^v)}_{\subseteq D_q^v} \cup D_p^2 \cup A \cup F = P$$

where $A_1 \cup A_2 \subseteq A$ and $F_2 \subseteq F$. Hence we get $D_p^2 \in \mathcal{Z}_A$ (since $p$ ignores the king), $D_q^v \in \mathcal{Z}_A$ (since $v_q \neq v$) and $(A, F) \in \mathcal{Z}$ in contradiction to $Q(P, \mathcal{Z})$. □

**Lemma 10 (Stop-Implication).** *If a correct player stops early then every correct player already prefers the same value $v$ and will do so during every subsequent communication round.*

*Proof.* Consider the first iteration of the for-loop in which some correct players stop. Let $p$ be such a player and $v = v_p \neq 2$ be his preferred value. Player $p$'s stopping implies that the king's value is ignored by $p$ and hence every correct player $q$ prefers the same value $v_q = v$ by Lemma 9. Due to Lemma 6, $D_q^2 \subseteq P \setminus D_p^v$ holds and since $P \setminus D_p^v \in \mathcal{Z}_A$ by the stop condition for player $p$ we immediately get $D_q^2 \in \mathcal{Z}_A$. Hence every correct player $q$ ignores the king's value and $v_p = v_q = v$ at the end of the loop. By Lemma 8 agreement on this value persists for every further communication round. □

**Lemma 11 (Consistence).** *At the end of any for-loop with a correct king all correct players prefer the same value.*

*Proof.* If any correct player has stopped so far then consistence follows by the Lemmas 8 and 10. Hence suppose that no correct player has stopped so far, and suppose some $k^{th}$ iteration of the for-loop with $p_k$ being correct. If all correct players replace their values $v := \min(1, w)$ we are done. Suppose now that at least one correct player $p$ ignores the value sent by the king. Hence, by Lemma 9, every correct player prefers the same value $v_p$ before Communication Phase 3. Hence especially the king prefers $v_p$ and every correct player who replaces his value replaces it with $v_p$. □

**Lemma 12.** *Let $C$ be the set of players that are actively or fail-corrupted. Agreement Protocol 2 achieves agreement and all correct players terminate the protocol after at most $|C| + 2$ iterations of the for-loop.*

*Proof.* That Agreement Protocol 2 achieves agreement follows immediately by the Lemmas 8 and 11, and the fact that there is at least one iteration of the for-loop with a correct king. It remains to show that, at the end of the first loop which is entered by all correct players with the same value $v \neq 2$, all correct players have stopped with value $v$. Suppose that all correct players enter some loop with the same value $v$. Due to the persistence property of `MakeUnique` they also enter `Unicast` with this value and hence, due to Lemma 7, they still hold this value after `Unicast` and $P \setminus D^v \in \mathcal{Z}_A$. Hence every correct player stops according to Line 7 of the protocol. □

### 4.5 Optimizations

Agreement Protocol 2 can be optimized in the following ways.

I. Depending on the concrete adversary structure $\mathcal{Z}$ the for-loop does not necessarily have to run over all $n$ possible kings since it is only required that at least one of the kings be correct.
II. Every correct player may stop the protocol immediately after the loop in which he plays the king because all correct players will start the next loop with his value.

III. In order to save one communication round in each loop, Communication Phase 3 (i.e. king's value distribution) can be integrated into `Unicast` by the king already computing his distribution value in advance after the `SendToAll` round of `Unicast`:
   1. $\widetilde{D}^i := \{ p_l \in P \mid R^l = i \}$ for $i \in \{0, 1, 2\}$;
   2. $\widetilde{v} := \begin{cases} 0, \text{ if } \widetilde{D}^0 \notin \mathcal{Z}_A \\ 1, \text{ if } \widetilde{D}^1 \notin \mathcal{Z}_A \\ 2, \text{ else.} \end{cases}$

   This value $\widetilde{v}$ can be sent by the king $p_k$ already during the `MakeUnique` round of `Unicast` without harming the protocol's correctness. In order to see this suppose the king to be correct.
   – If all correct players consider the king then they all prefer the same value at the end of this loop. The king is only considered if agreement did not hold at the beginning of the loop and hence it does not matter which value is sent by the king.
   – If at least one correct player ignores the king then for some $v \in \{0, 1\}$ $D_p^v \notin \mathcal{Z}_A$ holds for every correct player $p$ by Lemma 9. But since $D_{p_k}^v \subseteq \widetilde{D}_{p_k}^v$ this is the value $\widetilde{v} = v$ that is sent by $p_k$.

**Theorem 3.** *For any player set $P$ and adversary structure $\mathcal{Z}$ satisfying $Q(P, \mathcal{Z})$, Agreement Protocol 2 (by including the optimizations of this section) reaches agreement. Let $C$ be the set of players that actually misbehave in the protocol (by failing or sending false values), then all correct players terminate the protocol after at most $3(|C| + 2)$ communication rounds.*

*Proof.* The theorem follows by Lemma 12 and Optimization III of this section. □

### 4.6 Comparison with Previous Results

Agreement Protocol 2 (with optimizations) can be applied to the threshold model in [GP92] as well as to the general active adversary model in [FM98].

The protocols of [GP92] for the threshold model with actively and fail-corrupted players involve $5(t+1)$ communication rounds in the worst case. Provided that only some $c \leq b$ players are actually corrupted, then only $5(c+2)$ communication rounds are needed (whereas early stopping is not proven for $b < c < t$). Our improvement of these protocols is two-fold. First, the worst case round complexity is only $3(t+1)$. Second, we achieve early stopping independently of any additional constraint on the number $c$ of actually corrupted players, i.e., provided that some $c < t$ players are actually corrupted, the round complexity is at most $3(c+2)$.

In the general adversary model of [FM98] with only active player corruption the tight bound for broadcast and agreement to be achievable is that no three adversary sets $A_i \in \mathcal{Z}_A$ $(i \in \{1, 2, 3\})$ cover the player set $P$. This implies that there is at least one player set $S \notin \mathcal{Z}_A$ of cardinality $|S| \leq \lceil \frac{n}{3} \rceil$ since otherwise this condition would be violated. According to Optimization I, it therefore suffices to define the for-loop over the set $S$ since this set contains at least one correct player. Hence Agreement Protocol 2 involves at most $3\lceil \frac{n}{3} \rceil \leq n + 2$ communication rounds. Provided that $c$ players are actually corrupted then only

$\min(3(c+2), 3\lceil \frac{n}{3} \rceil)$ communication rounds are needed. In contrast to these results the protocols of [FM98] need $2n$ communication rounds in order to achieve polynomial communication complexity.

## 5 Acknowledgments

The authors would like to thank Ronald Cramer, Juan Garay and Martin Hirt for helpful comments and interesting discussions.

## References

[BDDS87] A. Bar-Noy, D. Dolev, C. Dwork, and H. R. 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*, pp. 42–51, Vancouver, British Columbia, Canada, 10–12 Aug. 1987.

[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.

[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.

[FM88] P. Feldman and S. Micali. Optimal algorithms for Byzantine agreement. In *Proc. 20th ACM Symposium on the Theory of Computing (STOC)*, pp. 148–161, 1988.

[FM98] M. Fitzi and U. Maurer. Efficient Byzantine agreement secure against general adversaries. In *Distributed Computing — DISC*, volume 1499, pp. 134–148, 1998.

[GM93] J. A. Garay and Y. Moses. Fully polynomial Byzantine agreement in $t+1$ rounds (extended abstract). In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pp. 31–41, San Diego, California, 16–18 May 1993.

[GP92] J. A. Garay and K. J. Perry. A continuum of failure models for distributed computing. In A. Segall and S. Zaks, editors, *Distributed Algorithms, 6th International Workshop, WDAG '92*, volume 647 of *Lecture Notes in Computer Science*, pp. 153–165, Haifa, Israel, 2–4 Nov. 1992. Springer.

[HM97] M. Hirt and U. Maurer. Complete characterization of adversaries tolerable in secure multi-party computation. In *Proc. 16th ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 25–34, Aug. 1997.

[LF82] L. Lamport and M. J. Fischer. Byzantine generals and transaction commit protocols. Technical report, SRI International (Menlo Park CA), TR, 1982.

[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.

[PSL80] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, Apr. 1980.

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