Unconditional Byzantine Agreement and Multi-Party Computation Secure Against Dishonest Minorities from Scratch

Matthias Fitzi[1], Nicolas Gisin[2], Ueli Maurer[1], and Oliver von Rotz[1]

[1] ETH Zurich
Department of Computer Science
{fitzi,maurer}@inf.ethz.ch, ovonrotz@ergon.ch
[2] Geneva University
Group of Applied Physics
Nicolas.Gisin@physics.unige.ch

Abstract. It is well-known that n players, connected only by pairwise secure channels, can achieve unconditional broadcast if and only if the number t of cheaters satisfies $t < n/3$. In this paper, we show that this bound can be improved — at the sole price that the adversary can prevent successful completion of the protocol, but in which case all players will have agreement about this fact. Moreover, a first time slot during which the adversary forgets to cheat can be reliably detected and exploited in order to allow for future broadcasts with $t < n/2$. This even allows for secure multi-party computation with $t < n/2$ after the first detection of such a time slot.

Key words: Byzantine agreement, multi-party computation, unconditional security.

1 Introduction

1.1 Unconditional Broadcast and Multi-Party Computation

In this paper we consider a set P of n players. The goal is to achieve broadcast (or multi-party computation, in general), 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 of 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 fails is at most ε 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 ($\varepsilon = 0$).
The goal of broadcast is to have a sender consistently distribute some input value to all players.

Definition 1. *A protocol among n players such that one distinct player s (the sender) holds an input value $x_s \in \mathcal{D}$ (for some finite domain $\mathcal{D}$) and all players eventually decide on an output value in $\mathcal{D}$ is said to achieve* broadcast *(or* Byzantine agreement*) if the protocol guarantees that all correct players decide on the same output value $y \in \mathcal{D}$, and that $y = x_s$ whenever the sender is correct.*

Some weaker definition of broadcast will be important in the sequel of this paper.

Definition 2. *A protocol among n players such that one distinct player s (the sender) holds an input value $x_s \in \mathcal{D}$ and all players eventually decide on an output value in $\mathcal{D} \cup \{\perp\}$ (with $\perp \notin \mathcal{D}$) is said to achieve* weak broadcast *if the protocol guarantees the following conditions:*

- *If any correct player decides on some value $y \in \mathcal{D}$ then all correct players decide on a value in $\{y, \perp\}$.*[1]
- *If the sender is correct then all correct players decide on $y = x_s$.*

Note that broadcast for any finite domain can be achieved by combining broadcast protocols for a single bit ($\mathcal{D} = \{0, 1\}$) [TC84]. Hence, for the constructions in the sequel, we will focus on protocols for binary broadcast.

Broadcast is a special case of the more general problem of secure *multi-party computation* (*MPC*) where the players want to distributedly evaluate some agreed function on their inputs in a way preserving privacy of the players' inputs and correctness of the computed result.

1.2 Previous Work

Broadcast: For the standard communication model with a complete synchronous network of pairwise authenticated channels, Pease, Shostak, and Lamport [PSL80] 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 a network of secure channels and unconditional security, i.e., when even allowing a negligible error probability, as proven by Karlin and Yao [KY]. The first optimally resilient protocol that is efficient was proposed by Dolev et al. [DFF$^+$82]. For the case that broadcast among every subset of three players is possible (in contrast to the standard model with only pairwise communication), Fitzi and Maurer [FM00] proved that (global) broadcast is possible if and only if $t < n/2$. In another line of research, Baum-Waidner, Pfitzmann, and Waidner [BPW91,PW92] proved that broadcast during some precomputation stage allows to later achieve broadcast that tolerates any number of corrupted players ($t < n$), i.e., that the functionality of the prior broadcast can be preserved for any later time.

Multi-party computation: The concept of general multi-party computation (MPC) was introduced by Yao [Yao82] with a first complete solution given by

[1] That is, interpreting $\perp$ as "invalid", this condition expresses that no two correct players may decide on valid values that are distinct.

Goldreich, Micali, and Wigderson [GMW87] — though with computational security. Ben-Or, Goldwasser, and Wigderson [BGW88], and, Chaum, Crépeau, and Damgård [CCD88], proved that, in the standard model with pairwise secure channels, *unconditionally* secure MPC is achievable if and only if $t < n/3$ by giving efficient protocols for the achievable cases. Beaver [Bea89], and independently, Rabin and Ben-Or [RB89] later proved that, when additionally given global broadcast among the players, unconditionally secure MPC is achievable if and only if $t < n/2$ (see also Cramer et al. [CDD$^+$99]). The result in [FM00] hence implies that broadcast among three players (i.e., 2-*cast*) is sufficient in order to achieve MPC for $t < n/2$.

1.3 Contributions

In this paper we investigate how the bound $t < n/3$ for the achievability of broadcast can be improved. Obviously, this requires a modification of the model.

In a first model, additionally to pairwise authenticated communication channels, we assume the existence of an external information source that distributes correlated random variables to the players that correspond to some simple probability distribution. We show that in this model broadcast and MPC are achievable for $t < n/2$.

Our second model assumes only standard communication, namely pairwise secure communication channels, but the goal is to achieve a slightly weaker form of broadcast or MPC, called *detectable broadcast* (or *MPC*), where the adversary can force abortion of the protocol but in which case all correct players have agreement about this fact, i.e., they commonly detect that the protocol was not successful. Besides this, the adversary can neither violate correctness, nor privacy, nor any other condition of the original problem. In other words, the detectable variant of a problem can be seen as the original problem without requiring robustness.

We show that detectable broadcast and MPC are achievable for $t < n/2$ by presenting efficient protocols that are based on the protocols given for the model involving an external information source. Consider, for example, the special case of $n = 3$ and $t = 1$ where broadcast and MPC are not achievable. Our results imply that such a protocol can nevertheless be run in an "optimistic" manner. If no player is corrupted then the protocol satisfies all conditions of the original problem whereas one corrupted player can only make the protocol abort in the worst case. This is strictly more than previously achievable.

Furthermore, a slightly modified version of the protocol achieves that a first time slot where the adversary is not actively cheating can be detected and exploited in order to establish (standard) broadcast for the future. This could be seen as a *lunch-time attack against the adversary*: if once the adversary is absent for a short period[2], the players can secure themselves for future broadcast that will be reliable even when the adversary will be present again. Together with the

[2] This could for instance be enforced by rebooting one or more of the servers.

result of [Bea89,RB89,CDD$^+$99] this more generally allows for future MPC for $t < n/2$ as soon as such a period has been detected.
As opposed to the results in the model with an external information source or the results in [Bea89,RB89,CDD$^+$99,FM00,BPW91,PW92], in this model, the bound of $t < n/2$ is achieved *from scratch*, i.e., with no further assumptions on the communication model than pairwise secure channels.
Our "optimistic" model is of particular interest when faults or corruption are expected to be rare but to appear in bursts. Virus infections of servers, for example, only occur from time to time but then it must be expected that many servers get infected at the same time. A first phase where no server is infected can thus be exploited in order to "vaccinate" the system against future infections of any minority of the servers.

2 Summary of Required Previous Results

In this section we briefly summarize previous results that are important for the results derived in this paper.

Proposition 1. *[Bea89,RB89,CDD$^+$99] Consider a set of n players. In the communication model with a complete, synchronous network of pairwise secure channels among the players and global broadcast channels unconditionally secure MPC is (efficiently) achievable if and only if at most $t < n/2$ players are actively corrupted.*

Proposition 2. *[FM00] Consider a set of n players. In the communication model with a complete, synchronous network of pairwise authenticated channels among the players and broadcast among each set of three players (i.e., 2-cast) unconditionally secure global broadcast is (efficiently) achievable if and only if at most $t < n/2$ players are actively corrupted.*

Finally, an unconditionally secure protocol for weak 2-cast can be easily turned into an unconditionally secure protocol for 2-cast. Consider a sender s and two recipients r_0 and r_1. Then the following protocol `Amplify` achieves 2-cast.

Protocol 1: `Amplify` [Precondition: s, r_0 and r_1 have executed weak 2-cast]

1. s decides on his own input to the prior weak 2-cast;
2. r_0, r_1: exchange decision values y_0 and y_1;
3. r_k ($k \in \{0,1\}$): adopt other recipient's decision value if and only if $y_k = \perp$;
4. r_k ($k \in \{0,1\}$): **if** $y_k = \perp$ **then** $y_k = 0$ **fi**;

Note that resilience of this 2-cast protocol against one single corrupted player immediately implies resilience against any number of corrupted players since, in the presence of more than one corrupted player, the only condition to be satisfied is that a correct sender decides on his own input value — which is obviously guaranteed.

Proposition 3. *[FM00] Consider a set of* 3 *players. In the model with a complete, synchronous network of pairwise authenticated channels weak* 2*-cast (i.e., weak broadcast) unconditionally secure against one corrupted player implies efficient* 2*-cast unconditionally secure against any number of corrupted players.*

Hence, as follows from the previous propositions, in order to show that global broadcast or MPC are efficiently achievable for $t < n/2$, it is sufficient to show that the simulation of weak 2-cast among any set of three players is possible that is unconditionally secure against one corrupted player. This fact is captured by the following proposition.

Proposition 4. *Consider a set of n players.*
In the model with a complete, synchronous network of pairwise authenticated *channels weak* 2*-cast unconditionally secure against one actively corrupted player implies efficient* broadcast *unconditionally secure against $t < n/2$ actively corrupted players.*
In the model with a complete, synchronous network of pairwise secure *channels weak* 2*-cast unconditionally secure against one actively corrupted player implies efficient* MPC *unconditionally secure against $t < n/2$ actively corrupted players.*

3 Broadcast and MPC with External Information

It can be easily proven that an additional global random source (i.e., a beacon) does not help to improve the classical bound of $t < n/3$ for broadcast (and hence for MPC in general) in the standard model by extending the proofs in [FLM86,FGMO01]. However, by slightly modifying the functionality of such a beacon, as described in the following section, it does. This functionality will be exploited in order to simulate broadcast for $t < n/2$ and hence allowing for general MPC with respect to the same bound.[3]

3.1 The Q-Flip Model

We assume the standard communication model with a complete (fully connected) synchronous network of *pairwise authenticated channels* among the players. But similarly to the model in [FM00] we assume some additional primitive among each triple of players, called *Q-Flip*.
Q-Flip, as described for the three players p_0, p_1, and p_2, is a random generator that (for every invocation), with uniform distribution, generates a random permutation on the elements $\{0, 1, 2\}$, $(x_0, x_1, x_2) \in \{(0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), (2, 1, 0)\}$, and sends the element x_i ($i \in \{0, 1, 2\}$) to player p_i.

[3] Note that a straightforward solution could be achieved with the help of an external information source that simulates the whole protocol in [BPW91]. However, this would be a rather complex task to be performed by an information source requiring a lot of mathematical structure. In contrast, our solution is based on very simple correlated information.

No single player p_i learns more about the permutation than the value x_i which he receives, i.e., a single player does not learn how the remaining two values are assigned to the other players.
The Q-Flip primitive helps to build pairwise one-time pads between the players (see Appendix A) and hence the authenticated channels can be easily turned into *secure channels*. This allows us to assume secure pairwise communication for the rest of this section.
The Q-Flip primitive was originally motivated by quantum entanglement considerations about the Byzantine agreement problem. A detailed description of the quantum physical aspects of our results is given in [FGM01].

3.2 Broadcast for $\mathbf{t < n/2}$

Since weak 2-cast secure against one corrupted player implies efficient global broadcast for $t < n/2$ (Proposition 4), it is sufficient to demonstrate the existence of an efficient protocol for weak 2-cast that tolerates one corrupted player. We now describe such a protocol for a sender s and two recipients r_0 and r_1.[4]

Let $x_s \in \{0, 1\}$ be the input of sender s, and y_0 and y_1 be the values the players r_0 and r_1 finally decide on (whereas s always implicitly decides on $y_s = x_s$). The primitive Q-Flip is invoked some m times and each player receives a sequence of m elements in $\{0, 1, 2\}$, i.e., s receives $Q_s = (Q_s[1], \ldots, Q_s[m])$, r_0 receives $Q_0 = (Q_0[1], \ldots, Q_0[m])$, and r_1 receives $Q_1 = (Q_1[1], \ldots, Q_1[m])$, where the triplet $(Q_s[i], Q_0[i], Q_1[i])$ represents the outcome of the i-th invocation of Q-Flip. The protocol now proceeds as follows:
First, s sends to r_0 and r_1 his input bit x_s and the set σ_s of all indices $i \in \{1, \ldots, m\}$ such that s received the complement of x_s for the i-th invocation of Q-Flip (see Figure 1–A):

$$\sigma_s = \{i \in \{1, \ldots, m\} : Q_s[i] = 1 - x_s\} \ . \tag{1}$$

If this is done correctly then σ_s is of large size (i.e., approximately $m/3$, which is important for good statistics on the corresponding Q-Flips) and both recipients r_k ($k \in \{0, 1\}$) never received the value $1 - x_s$ for any Q-Flip invocation with respect to σ_s: $\{i \in \sigma_s : Q_k[i] = 1 - x_s\} = \emptyset$.
Let x_k and σ_k ($k \in \{0, 1\}$) be the information that (potentially faulty) s actually sent to recipient r_k. The recipients now decide on $y_k = x_k$ if and only if σ_k is of large size and the value $1 - x_k$ was never received with respect to σ_k:

$$y_k := \begin{cases} x_k \text{ , if } \ (\sigma_k \text{ large}) \ \wedge \ (\{i \in \sigma_k : Q_k[i] = 1 - x_k\} = \emptyset) \\ \perp \ \text{ , else .} \end{cases} \tag{2}$$

Now, r_0 sends to r_1 his value y_0 and the set ρ_0 of all indices $i \in \sigma_0$ such that he received x_0 for the corresponding Q-Flip invocation:

$$\rho_0 = \{i \in \sigma_0 : Q_0[i] = x_0\} \ . \tag{3}$$

[4] The protocol descriptions in this paper do not explicitly care about received values that are outside a domain. We implicitly assume that any value received outside some expected domain $\mathcal{D}$ is automatically replaced by an arbitrary value inside $\mathcal{D}$.

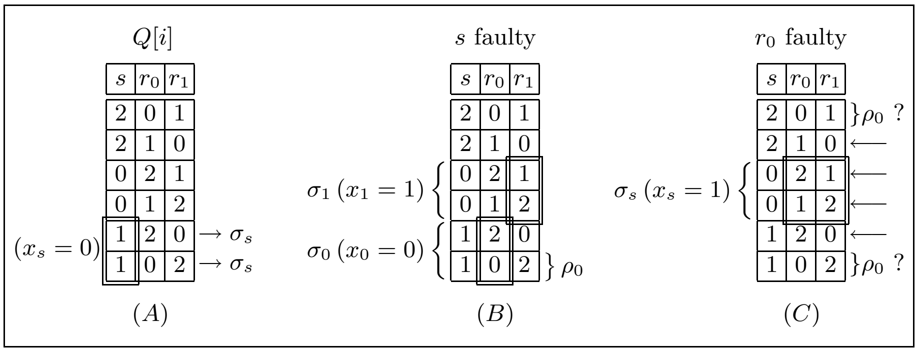

Fig. 1. *(A) Possible outcomes of Q-Flip and selection of σ_s; (B,C) Basic cheating strategies for s and r_0.*

If $y_0 \neq y_1$ but $y_0, y_1 \in \{0,1\}$, y_1 now redecides in the following way:

$$y_1 := \begin{cases} y_0 \text{ , if } \; (\rho_0 \text{ large}) \;\wedge\; (\{i \in \rho_0 \setminus \sigma_1 : Q_1[i] = 2\} \overset{\text{almost}}{=} \rho_0) \\ y_1 \text{ , else .} \end{cases} \tag{4}$$

We give a rough argumentation why none of the players can make the protocol fail.

$\mathbf{r_1}$ faulty: r_1 cannot significantly misbehave since r_1 is silent.

s faulty: In order to make the protocol fail, s must achieve that r_0 and r_1 decide on distinct values $y_0 \neq y_1$ such that $y_0, y_1 \in \{0,1\}$ (Equation (2)) and that r_1 does not redecide on $y_1 = y_0$ according to Equation (4). The only way to achieve this is to select $x_0 \in \{0,1\}$ and $x_1 = 1 - x_0$, and to basically compose large sets σ_0 and σ_1 as shown in Figure 1–B (as shown with respect to $x_0 = 0$).[5] But then, r_0 learns a large set ρ_0 of indices i (Figure 1–B, last row) such that, mainly, $i \notin \sigma_1$ and $Q_1[i] = 2$ which will "convince" r_1 to redecide to $y_1 = y_0$ according to Equation (4).

$\mathbf{r_0}$ faulty: In order to make the protocol fail, r_0 must achieve that r_1 redecides on $y_1 = y_0 \neq x_s$ according to Equation (4). Since correct s sends $\sigma_s = \{i \in \{1,\ldots,m\} : Q_s[i] = x_s\}$ to both players, r_0 cannot come up with a large set ρ_0 of indices i such that most of them satisfy $i \notin \sigma_s$ and $Q_1[i] = 2$ (see Figure 1–C) since r_0 cannot distinguish between the outcomes corresponding to the first and the last row.[6]

The detailed protocol is described by Protocol `Weak 2-Cast`. There, two free protocol parameters are introduced, m_0 ($m_0 = \Omega(m)$; $m_0 < m/6$) for asserting that the sets σ_s and ρ_0 are of sufficiently large cardinality, and λ ($\frac{1}{2} < \lambda < 1$) for

[5] Note that for every selection $i \in \sigma_k$ ($k \in \{0,1\}$) such that $Q_s[i] \neq 1 - x_k$, it holds with a probability of $\frac{1}{2}$ that $Q_k[i] = 1 - x_k$, which makes r_k decide on $\perp$ according to Equation (2).

[6] Note, that r_0 completely learns all instances indicated by an arrow but nothing more about the other instances.

Protocol 2: `Weak 2-Cast` [s to send $x_s \in \{0,1\}$ to r_k ($k \in \{0,1\}$)]

0. s, r_0, r_1: invoke primitive Q-Flip m times;

1. $s \xrightarrow{\text{send}} r_0, r_1$: x_s, $\sigma_s = \{i \in \{1,\ldots,m\} : Q_s[i] = 1 - x_s\}$; [$r_k$ receives x_k, σ_k]
   s: $y_s := x_s$;

2. r_k: `if` $(|\{i \in \sigma_k : Q_k[i] = x_k\}| \geq m_0) \wedge (\{i \in \sigma_k : Q_k[i] = 1 - x_k\} = \emptyset)$ `then`
   $y_k := x_k$
   `else` $y_k := \perp$ `fi`;

3. $r_0 \xrightarrow{\text{send}} r_1$: y_0, $\rho_0 = \{i \in \sigma_0 : Q_0[i] = y_0\}$; [$r_1$ receives y_{01} and ρ_{01}]

4. r_1: `if` $(\perp \neq y_{01} \neq y_1 \neq \perp) \wedge (|\rho_{01}| \geq m_0) \wedge$
   $(|\{i \in \rho_{01} \setminus \sigma_1 : Q_1[i] = 2\}| \geq \lambda\,|\rho_{01}|)$ `then`
   $y_1 := y_{01}$
   `fi`;

the test according to Equation (4). Both parameters will be fixed for the final analysis of the protocol, which is given in Appendix C. The analysis yields:

Theorem 1. *In the Q-Flip model, unconditionally secure broadcast and MPC are efficiently achievable for $t < n/2$.*

Proof. The theorem follows from Lemma 9 and Proposition 4. □

The following observation about the given protocol is important for the construction of our protocol for the model in Section 4.

Observation 1: In Protocol `Weak 2-Cast`, player r_1 does not send a single message but only participates in a passive way. Even when turning the protocol into 2-cast (by appending Protocol `Amplify`) the single message sent by r_1 is only considered by r_0 if r_0 has already reliably detected the sender s to be faulty.

4 Detectable Broadcast and MPC for t < n/2

We assume the standard communication model with a complete (fully connected) synchronous network of *pairwise secure channels* among the players. Based on the solution for the model in the previous section, we present a protocol for detectable broadcast and MPC among n players that tolerates $t < n/2$ corrupted players.

Definition 3. *A protocol among n players is said to achieve* detectable broadcast *if it satisfies the following conditions:*

- **Correctness:** *If at most $t < n/2$ players are corrupted during the protocol then all correct players commonly accept or commonly reject the protocol. If they accept then the protocol achieves broadcast.*

– **Robustness:** *If no player is corrupted during the protocol then all players accept.*
– **Fairness:** *If any correct player rejects the protocol then the adversary gets no information about the sender's input.*

Note the difference between weak broadcast and detectable broadcast. Weak broadcast only guarantees that no two correct players decide on distinct values inside the domain $\mathcal{D}$. But generally, no player detects whether there are still any correct players that decided on $\perp \notin \mathcal{D}$. In contrast, at the end of detectable broadcast, all correct players agree on whether or not broadcast has been achieved.
Along the lines of [BPW91,PW92], detectable broadcast can be extended to a precomputation protocol that, when once successfully completed, will allow for future broadcast secure against $t < n/2$ corrupted players.

Definition 4. *A protocol among n players is said to achieve* detectable precomputation for broadcast *(or* MPC*, respectively) if it satisfies the following conditions:*

– **Correctness:** *If at most $t < n/2$ players are corrupted during the protocol then all correct players commonly accept or commonly reject the protocol. If they accept then, after the protocol, broadcast (or MPC, respectively) for $t < n/2$ will be achievable.*
– **Robustness:** *If no player is corrupted during the protocol then all players accept.*
– **Independence:** *A correct player's intended input value for any precomputed broadcast (or MPC, respectively) is statistically independent from the information he has sent during the precomputation.*

Independence implies, first, that a correct sender is not required to already know his input for any future protocol during the precomputation, and second, that the adversary gets no information about any future inputs by correct senders.
As opposed to detectable broadcast, the advantage of detectable precomputation for broadcast is that the preparation is separated from the actual execution of the broadcasts, i.e., only the precomputation is "detectable". As soon as the precomputation has been successfully completed standard broadcast with resilience $t < n/2$ is achievable. By applying many detectable precomputations in parallel, as many broadcasts can be precomputed for as will be later required. By using the techniques in [PW92], in order to be able to perform u later broadcasts, the number of broadcasts to be detectably computed for in advance can be made as low as polynomial in n and logarithmic in u.

We now proceed by presenting according protocols for the special case $n = 3$ and $t = 1$. These solutions can then be extended to any n and $t < n/2$ based on Proposition 4.

4.1 Detectable Precomputation for 2-Cast

The idea of our precomputation is to have recipient r_1 (the silent recipient of Protocol `Weak 2-Cast`) take the part of the Q-Flip source to correctly distribute sequences Q, and to base future 2-cast on Protocol `Weak 2-Cast` of Section 3 using Q. Since r_1 is silent during Protocol `Weak 2-Cast`, he will not be able to exploit his complete knowledge about Q in order to mislead anybody else but will be able to completely check consistency of information about Q provided by any other player. In the case that r_1 distributes this information correctly, this setup will allow for some (limited) number of 2-casts by s (and even by r_0 — by having s and r_0 switch their roles).
Clearly, we cannot guarantee that r_1 performs this task correctly. But by cross-checking, s and r_0 will be able to detect any inconsistency in his information that would enable the adversary to make a future 2-cast protocol (based on this information) fail with any reasonable chance.
Finally, any player who has successfully completed this setup procedure will be convinced that the Q-Flip information by r_1 is indeed (mostly) consistent (at least) among all correct players: r_1 since he made up the states himself, and s and r_0 because the cross-check was successful.
However, this does not necessarily imply that the correct players agree on the precomputation outcome. The adversary can still enforce that some correct player successfully completes (i.e., accepts) whereas the other one rejects the outcome.
Finally, in order to achieve agreement on the precomputation outcome, we have s and r_0 "2-cast" whether they accept using instances of the 2-cast protocol the precomputation is actually preparing for but such that enough Q-Flip data remains for one or more 2-casts (i.e., for the later 2-cast(s) we are in fact preparing for). Note that these "2-casts" are not necessarily reliable, but:

Observation 2: (Note that we can assume that $t = 1$, see Proposition 3)
If r_1 distributes inconsistent Q-Flip information then all correct players reject already before the invocation of the two final 2-cast protocols. Hence, if any correct player still accepts just before the invocation of the two final 2-cast protocols then the Q-Flip information is consistent and hence the those protocols achieve 2-cast.

Hence, if either s or r_0 "2-casts" his rejection at the end of the precomputation then all correct players that are not yet rejecting redecide to reject (since they know that the final 2-cast protocols are reliable). On the other hand, if the final 2-cast protocols are unreliable, then all correct players have already decided to reject before, and they simply ignore whatever is communicated during the last round.
We now proceed with a more detailed description of the protocol in three steps. Protocol `VerifSetup` describes how r_1 prepares and distributes Q and how s and r_0 cross-check their information gotten from r_1. Protocol `Conditional 2-Cast` describes an algorithm for 2-cast based on sequences Q as generated by player r_1. Finally, these two protocols are combined in order to obtain our final protocol for detectable precomputation, Protocol `OptPrecomp`.

Verifiable Setup of Q-Flip Information A straightforward way to get a verifiably correct setup Q would be a cut-and-choose manner where r_1 prepares more information than finally required of which s and r_0 would select a random part for testing which then would be discarded whereas the remaining part would be kept. The problem of this approach is that also r_1 (who sets up the states) must learn which part has been discarded — information that would have to be distributed by broadcast, which is not yet available.
As a consequence, in the following protocol, yet some partial information is tested by s and r_0 but is not discarded. It can be shown that the information thus leaked to s and r_0 will not help anyone of them in order to disrupt the final 2-cast based on this setup.

Protocol 3: `VerifSetup` [s, r_0, r_1 to prepare shared Q-Flip Information]

0. All players start with status ACCEPT.
1. r_1 sets up $m = 6\mu$ Q-Flip states each such that each possible outcome occurs exactly μ times; randomly permutes the states, and secretly sends the according values to s and r_0.
2. s and r_0 locally check whether they hold exactly $\frac{m}{3} = 2\mu$ of each value $(0, 1, 2)$. Anybody whose check fails changes to status REJECT.
3. r_0 sends to s his status (ACCEPT or REJECT) and, if he accepts, $\tau \ll m$ random indices in $\{1, \ldots, m\}$ and his according values.
4. s checks whether all those values from r_0 differ from his own values for the according states. If r_0 sent REJECT or if the check fails (i.e., reveals collisions) then s sets his status to REJECT.
5. s tells r_0 his status.
6. If s sent REJECT then r_0 sets his status to REJECT.

Notice that r_1 can undetectedly misbehave by distributing the 6 possible states with non-uniform cardinality, e.g., by repeatedly setting up only the three states determined by $(Q_s[i] = 0, Q_0[i] = 1)$, $(Q_s[i] = 1, Q_0[i] = 2)$, and $(Q_s[i] = 2, Q_0[i] = 0)$ — which makes s and r_0 nevertheless accept according to steps 2 and 4. However, regarding players s and r_0 being correct we shall see that, for the final 2-cast protocol, we only require that each player holds 2μ of each value in $\{0, 1, 2\}$ and that there are (almost) no collisions $Q_s[i] = Q_0[i]$.

Conditional 2-Cast Conditional 2-cast is a protocol based on one instance of protocol `VerifSetup`. The goal of this protocol is that, given that any correct player accepted the setup protocol, the protocol achieves 2-cast with overwhelming probability. Note that besides its termination we do not require anything for this protocol for the case that all correct players rejected the setup protocol.
The protocol is described by Protocol `Conditional 2-Cast`, which is basically still the same as for the model of Section 3 (i.e., Protocol `Weak 2-Cast` followed by Protocol `Amplify`). We give a short motivation for the changes compared to the former protocol.

– Step 0 drops out since the Q-Flips are provided by Protocol `VerifSetup`.
– Step 1: r_1 has full knowledge of Q — hence only x_s is sent to r_1.
– Step 2: We exploit that exactly 2μ of each value must be prepared (as is required by Protocol `VerifSetup`) in order to enforce good statistical properties of σ_0. Note that the condition $(|\{i \in \sigma_0 : Q_0[i] = x_0\}| \geq m_0)$ must not be checked anymore because faulty r_1 could have undetectedly avoided such states during setup. The only thing to be checked is that s claims no collisions — which is weakened to allow ε_0 collisions because r_1 undetectedly could have set up a small portion of such errors.
– Step 3: r_0 sends his original data from s since r_1 can recalculate everything.

Protocol 4: `Conditional 2-Cast` [s to distribute $x_s \in \{0,1\}$ to r_k $(k \in \{0,1\})$]

1. $s \xrightarrow{\text{send}} r_0, r_1$: x_s; [r_k $(k \in \{0,1\})$ receives x_k]
   $s \xrightarrow{\text{send}} r_0$: $\sigma_s = \{i \in \{1,\ldots,m\} : Q_s[i] = 1 - x_s\}$; [$r_0$ receives σ_0]
   s: $y_s := x_s$;

2. r_0: **if** $(|\sigma_0| = 2\mu)$ $\wedge$ $(|\{i \in \sigma_0 : Q_0[i] = 1 - x_0\}| < \varepsilon_0)$ **then**
   $y_0 := x_0$
   else $y_0 :=\perp$ **fi**;
   r_1: $y_1 := x_1$;

3. $r_0 \xrightarrow{\text{send}} r_1$: x_0, σ_0; [r_1 receives x_{01} and σ_{01}]

4. r_1: determine y_{01} as r_0 determines y_0 according to step 2;
   $\rho_{01} := \{i \in \sigma_{01} : Q_0[i] = y_{01}\}$;
   if $(\perp \neq y_{01} \neq y_1 \neq \perp)$ $\wedge$ $(|\rho_{01}| \geq m_0)$ $\wedge$
   $(|\{i \in \rho_{01} : Q_1[i] = 2\}| \geq \lambda\,|\rho_{01}|)$ **then**
   $y_1 := y_{01}$
   fi;
5. **Amplify**;

Definition 5. *Let $P_{\mathit{fail}|\mathit{acc}}$ be the probability that conditional 2-cast based on Q (as computed by the corresponding protocol* `VerifSetup`*) does not achieve 2-cast, given that one correct player (s, r_0, or r_1) accepted at the end of protocol* `VerifSetup`*.*

It remains to prove that $P_{\text{fail}|\text{acc}}$ can be made negligibly small (with the protocol remaining efficient). Our detailed proof would exceed the space limits of this paper and is hence omitted. We restrict ourselves to an informal (though plausible) argument:

– faulty s or r_0 cannot successfully misbehave since then r_1 is correct and sets up Q correctly; and only a small part of the states is revealed during the precomputation phase.

– r_1 cannot successfully misbehave since Q can only contain a small fraction of incorrect triplets (otherwise the precomputation would have failed), and since r_1 is silent.[7]

Lemma 1. *There exist parameters μ, $\tau < \mu$, ε_0, m_0, and λ for the protocols* `VerifSetup` *and* `Conditional 2-Cast` *such that the following conditions hold:*

– *If Protocol* `Conditional 2-Cast` *with the given parameters is based on a set Q that has been set up with Protocol* `VerifSetup` *with the given parameters which itself has been accepted by at least one correct player then the probability $P_{fail|acc}$ that the protocol does not achieve 2-cast is exponentially small in μ.*
– *Independently of Q, the protocol always terminates.*

Detectable Precomputation for b later 2-casts The final protocol for detectable precomputation is described in detail by Protocol `OptPrecomp`. It precomputes shared information for b later broadcast instances.

Protocol 5: `OptPrecomp` [s, r_0, r_1 to precompute for b 2-casts]

0. All players start in a status of ACCEPT.
1. Run $b+2$ executions of Protocol `VerifSetup` resulting in sets $S_1, \ldots, S_{b+2}$ of Q-Flip information.
2. s and r_0 decide on ACCEPT if and only if they accepted all $b+2$ executions of Protocol `VerifSetup` in step 1.
3. s and r_0 broadcast their status by applying Protocol `Conditional 2-Cast` with help of the sets S_{b+1} and S_{b+2}.
4. Anybody who received REJECT for one of the 2-casts sets his status to REJECT.

We now show that, whenever the probability $P_{\text{fail|acc}}$ is negligible, the correct players agree on their decision at the end of the precomputation and that, in case they accept, all b later 2-casts work correctly — both with overwhelming probability.

Lemma 2.

1. *At the end of protocol* `OptPrecomp` *the correct players will agree on their decision (ACCEPT or REJECT) with an error probability of at most $P_1 \leq 2P_{fail|acc}$. Moreover, if all players are correct, they will all agree on ACCEPT with error probability 0.*
2. *If the correct players accept the precomputation then, with an overall error probability of at most $P_2 \leq bP_{fail|acc}$, all b later 2-casts work reliably. Moreover, if all players are (and will be) correct, then all 2-casts work with error probability 0.*

[7] More precisely, the single message by r_1 that is sent during the amplification protocol in step 5 is only considered if r_1 is in fact correct.

Proof.

1. We distinguish two cases.
   (a) *At least one correct player accepts after step 2 of the protocol:* Then each conditional 2-cast in step 3 achieves 2-cast with an error probability of at most $P_{\text{fail}|\text{acc}}$ (see Definition 5). If now all correct players reject at the end of the protocol we are done. On the other hand, suppose that any correct player accepts at the end of the protocol. This player neither sent nor received a REJECT during step 3, and hence, with an error probability of at most $P_1 \leq 2P_{\text{fail}|\text{acc}}$, no other player sent or received a REJECT during step 3, and all correct players accept after step 4.
   (b) *All correct players reject after step 2:* Since no correct player ever changes from REJECT to ACCEPT, all correct players will reject at the end of the protocol.

   Finally, it can be easily seen by inspection that, in the case that all players are correct, all players agree on ACCEPT with error probability 0.
2. Any correct player that accepts at the end of the protocol already accepts after step 2 of the protocol and hence, by the definition of $P_{\text{fail}|\text{acc}}$, the overall error probability is at most $P_2 \leq bP_{\text{fail}|\text{acc}}$. Furthermore, if all players are (and will be) correct, none of the 2-cast protocols can fail. □

The full analysis of Protocol `OptPrecomp` together with Lemma 2 and Proposition 3 yields:

Lemma 3. *In the secure-channels model, efficient detectable precomputation for 2-cast among $n = 3$ players is achievable for any number of actively corrupted players ($t < n$).*

Together with Proposition 1 we get:

Corollary 1. *In the secure-channels model, efficient detectable precomputation for MPC among $n = 3$ players secure against one actively corrupted player ($t = 1$) is achievable.*

4.2 Detectable Precomputation for General n

In order to obtain a detectable precomputation protocol for general n and $t < n/2$, Protocol `OptPrecomp` can be applied in parallel for each triple of players and for every selection of a sender among them (i.e., $3\binom{n}{3}$ parallel protocols). We only need the following, minor changes:

1. The steps 5 of all parallel invocations of protocol `VerifSetup` are merged by having each of the n players tell each other whether or not he accepts with respect to all triples he is involved in, and to have each player reject in step 6 on any single reception of REJECT.

2. Protocol `Conditional 2-Cast` is turned into global conditional broadcast by applying the construction in [FM00].[8]
3. The steps 3 of all parallel invocations of protocol `OptPrecomp` are merged by having each player globally broadcast his status and by having the players decide to reject in step 4 on one single reception of REJECT for any one of the broadcasts.
4. In the final (conditional) broadcast protocol, each player of each a triple is involved as a sender in $2t$ different 2-casts with respect to this triple [FM00]. Furthermore, n instead of 2 conditional broadcasts are executed during step 3 of Protocol `OptPrecomp`. Hence, instead of $b+2$ invocations of `VerifSetup`, $2t(b+n)$ of them are required for each triple of players and for each selection of a sender among them.

The following Theorem immediately follows:

Theorem 2. *In the secure-channels model, efficient detectable precomputation for broadcast and MPC for n players and $t < n/2$ is achievable.*

Furthermore, detectable MPC can even be achieved in a slightly weaker model than with secure channels, namely in a model with classical authenticated channels and quantum channels. Since every player has the possibility to initiate rejection of the precomputation, we can apply quantum key agreement [BB84] in parallel to Protocol `OptPrecomp`. Whenever any quantum channel between two correct players would be eavesdropped they would detect it and just initiate rejection of the whole precomputation.

Theorem 3. *In the model with classical authenticated channels and quantum channels, efficient detectable precomputation for broadcast and MPC for n players and $t < n/2$ is achievable.*

5 Conclusions

To the best of our knowledge we have given the first examples of unconditionally secure protocols for broadcast and MPC in the secure channels model that tolerate $t < n/2$ active player corruptions from scratch, i.e., without any additional assumptions on the communication — at least in a way that the adversary cannot achieve anything more than make all players commonly abort the protocol. For the case of $n = 3$ players this is strictly more than previously achieved. For the case of general $n > 3$ it is achieved at the price that the adversary may cause non-completion only by corrupting one single player but requiring permanent corruption.

[8] Replacing every invocation of ordinary 2-cast by conditional 2-cast in their protocol immediately yields conditional global broadcast since a player who accepts after step 2 of the merged Protocol `OptPrecomp` knows that all triples involving correct players share correct Q-Flip information.

References

[BB84] C. H. Bennett and G. Brassard. An update on quantum cryptography. In *Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pp. 475–480. Springer-Verlag, 1985, 19–22 Aug. 1984.

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

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

[BPW91] B. Baum-Waidner, B. Pfitzmann, and M. Waidner. Unconditional byzantine agreement with good majority. In *Proceedings of STACS '91*, volume 480 of *LNCS*, pp. 285–295, Hamburg, Germany, 14–16 Feb. 1991. Springer.

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

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

[Chv79] V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25:285–287, 1979.

[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.

[FGM01] M. Fitzi, N. Gisin, and U. Maurer. Quantum solution to the byzantine agreement problem. To appear at Physical Review Letters, 87(21). Preliminary version: Quantum Physics, abstract quant-ph/0107127, 2001.

[FGMO01] M. Fitzi, J. A. Garay, U. Maurer, and R. Ostrovsky. Minimal complete primitives for unconditional multi-party computation. In *Proceedings of CRYPTO '01*, Lecture Notes in Computer Science, 2001.

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

[FM00] M. Fitzi and U. Maurer. From partial consistency to global broadcast. In *Proceedings of STOC '00*, pp. 494–503, Portland, Oregon, 2000. ACM.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game — a completeness theorem for protocols with honest majority. In *Proceedings of STOC '87*, pp. 218–229, 1987.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[KY] A. Karlin and A. C. Yao. Manuscript.

[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.

[PW92] B. Pfitzmann and M. Waidner. Unconditional byzantine agreement for any number of faulty processors. In *Proceedings of STACS '92*, volume 577 of *LNCS*, pp. 339–350. Springer, 1992.

[RB89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of STOC '89*, pp. 73–85, 1989.

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

[Yao82] A. C. Yao. Protocols for secure computations. In *Proceedings of FOCS '82*, pp. 160–164, 1982.

A Generation of One-Time Pads in the Q-Flip Model

Alice and Bob can generate a one-time pad (OTP) of length approximately k by using $3k$ Q-Flip invocations shared with an arbitrary third player Charlie and reporting to each other where they got a value different from 2. By this exchange of information Charlie does not get any additional information about the actual outcome of the Q-Flip invocations. Finally, the OTP is formed by those Q-Flip instances where both, Alice and Bob, got either 0 or 1, e.g., by Alice's according bits — which are the complements of Bob's.

B Chernoff and Hoeffding Bounds

For a detailed analysis of our protocols we will apply Chernoff and Hoeffding bounds [Hoe63,Chv79] in order to estimate upper bounds on their error probabilities.
The Chernoff bound gives an upper bound on the probability that of n independent Bernoulli trials the outcome deviates from the expected value by a given fraction.

Let X_i $(1 \leq i \leq n)$ be a sequence of independent random variables with expected value μ. By $\mathcal{C}(\mu, n, \lambda)$ we denote the Chernoff bound as follows

$$\begin{array}{l} \lambda < 1 : \mathcal{C}_{\downarrow}(\mu, n, \lambda) = \mathrm{Prob}(\sum_{i=1}^{n} X_i \leq \lambda\, \mu n) \leq e^{-\frac{\mu n}{2}(1-\lambda)^2} \\ \lambda > 1 : \mathcal{C}_{\uparrow}(\mu, n, \lambda) = \mathrm{Prob}(\sum_{i=1}^{n} X_i \geq \lambda\, \mu n) \leq e^{-\frac{\mu n}{3}(\lambda-1)^2} \end{array} \tag{5}$$

Furthermore, a bound by Hoeffding can be used to estimate tail probabilities of hypergeometric distributions. By the term $\mathcal{H}(N, K, n, k)$ we refer to a setting where N items are given of which K are "good". The experiment consists of selecting n out of the N items at random, and $\mathcal{H}(N, K, n, k)$ denotes the probability that at least k of the n selections are "good". Let $t = \frac{k}{n} - \frac{K}{N}$. The following inequation holds for any t such that $0 \leq t \leq 1 - \frac{K}{N}$ [Chv79],

$$\mathcal{H}(N, K, n, k) = \sum_{i=k}^{n} \binom{K}{i} \binom{N-K}{n-i} \binom{N}{n}^{-1} \leq e^{-2t^2 n} \ . \tag{6}$$

C Broadcast and MPC with an External Information Source: Details

Lemma 4. *Let $\lambda_0 < 1$ and $\lambda_1 > 1$. The probability P_{stats} that of m invocations of Q-Flip any one of the six possible outcomes in $\{(0,1,2), \ldots, (2,1,0)\}$ occurs either less than $m_0 = \lambda_0 \frac{m}{6} < \frac{m}{6}$ or more than $m_1 = \lambda_1 m > \frac{m}{6}$ times satisfies $P_{stats} \leq 6\, max(\mathcal{C}_{\downarrow}(\frac{1}{6}, m, \lambda_0), \mathcal{C}_{\uparrow}(\frac{1}{6}, m, \lambda_1))$.*[9]

[9] See Appendix B for the definition of $\mathcal{C}_{\downarrow}$ and $\mathcal{C}_{\uparrow}$.

Proof. The according probability for each particular outcome, e.g. $(0, 1, 2)$, can be independently estimated by the Chernoff bound (with random variable X_i representing the i-th Q-Flip). The overall probability is at most as large as the sum of these six probabilities (union bound). $\square$

Lemma 5 (All players correct). *If all players are correct and each possible outcome of Q-Flip appears at least m_0 times, then protocol* `Weak 2-Cast` *achieves weak 2-cast.*

Proof. By the given assumptions we have $\sigma_s = \sigma_0 = \sigma_1$ and $x_s = x_0 = x_1$ after step 1 of the protocol. Since $\sigma_s = \{i \in \{1, \ldots, m\} : Q_s[i] = 1 - x_s\}$, it holds for both recipients r_k that

$$\{i \in \sigma_k : Q_k[i] = 1 - x_k\} = \emptyset \quad \wedge \quad |\{i \in \sigma_k : Q_k[i] = x_k\}| \geq m_0 \ .$$

Hence $y_0 = y_1 = x_s$ after step 3 and hence also at the end of the protocol. $\square$

The following corollary follows from Observation 1 on page 6.

Corollary 2 ($\mathbf{r_1}$ possibly faulty). *If the players s and r_0 are correct and each possible outcome of Q-Flip appears at least m_0 times, then protocol* `Weak 2-Cast` *achieves weak 2-cast.*

It now remains to determine upper bounds on the error probability for the cases that s is faulty (P_s) or that r_0 is faulty (P_{r_0}). The following lemma will be used for the analysis of the former case in the proof of Lemma 7.

Lemma 6. *If each possible outcome of Q-Flip appears at least m_0 times and at most m_1 times and if (faulty) sender s submits $x_0 \in \{0, 1\}$ to r_0 and $x_1 = 1 - x_0$ to r_1 and selects k indices $i \in \{1, \ldots, m\}$ such that either*

- *$i \in \sigma_0 \cap \sigma_1$, or*
- *$i \in \sigma_0 \setminus \sigma_1$ such that $Q_s[i] = 2$*

then $y_0 = x_0$ and $y_1 = 1 - x_0$ (i.e., disagreement) holds at the end of the protocol with a probability of at most $(\frac{m_1}{m_0+m_1})^k$.

Proof. If s submits one single index $i \in \sigma_0 \cap \sigma_1$ then either $Q_0[i] = 1 - x_0$ or $Q_1[i] = 1 - x_1$ with a probability of at least $\frac{m_0}{m_0+m_1}$ (since the only information by s is $Q_s[i]$ and hence s risks, with the according probability, to produce a collision on either recipient's side which makes him decide on $\perp$).

On the other hand, if s submits one single index $i \in \sigma_0 \setminus \sigma_1$ such that $Q_s[i] = 2$ (the only possibility in order to achieve that $Q_0[i] = x_0$ and $Q_1[i] \neq 2$) then r_0 decides on $\perp$ with a probability of at least $\frac{m_0}{m_0+m_1}$ (since s risks, with the according probability, to produce a "collision" on r_0's side which makes him decide on $\perp$).

Finally, in order to achieve that $y_0 = x_0$ and $y_1 = 1 - x_0$ holds at the end of the protocol, s must prevent any single collision for all k index selections. This can be achieved with a probability of at most $(\frac{m_1}{m_0+m_1})^k$. $\square$

Lemma 7 (s possibly faulty). *If the players r_0 and r_1 are correct and each possible outcome of Q-Flip appears at least m_0 times and at most m_1 times, then protocol* `Weak 2-Cast` *fails to achieve weak 2-cast with probability at most* $P_s < \left(\frac{m_1}{m_0+m_1}\right)^{(1-\lambda)m_0}$.

Proof. The only way for s to make the protocol fail is to force the recipients to decide on distinct bits, i.e., $y_0 = x_0 = b \in \{0,1\}$ and $y_1 = x_1 = 1-b$. Hence both recipients must already decide on those values during step 3 of the protocol, which implies $|\rho_{01}| = |\rho_0| \geq m_0$ — since r_0 would set $y_0 = \perp$ otherwise. Furthermore, r_0 must not be able to convince r_1 to redecide on $y_1 = y_{01} = y_0$ during step 5 of the protocol. Since the first two conditions according to step 5 are satisfied, i.e.,

- $\perp \neq y_0 = y_{01} \neq y_1$ (since the recipients hold distinct bits), and
- $|\rho_{01}| \geq m_0$ (see above),

the last condition must be violated, i.e., it must hold that

$$|\{i \in \rho_{01} \setminus \sigma_1 : Q_1[i] = 2\}| < \lambda\,|\rho_{01}| \ .$$

Hence s must find some $\ell > (1-\lambda)\,|\rho_{01}| \geq (1-\lambda)m_0$ indices i such that either

- $i \in \rho_{01} \cap \sigma_1 \quad (\subseteq \sigma_0 \cap \sigma_1)$, *or*
- $i \in \rho_{01} \setminus \sigma_1 \quad \wedge \quad Q_1[i] \neq 2$ (and hence $Q_s[i] = 2$) ,

such that no collision occurs. By Lemma 6 this happens with probability at most $P_s < \left(\frac{m_1}{m_0+m_1}\right)^{(1-\lambda)m_0}$. □

Lemma 8 ($\mathbf{r_0}$ possibly faulty). *If the players s and r_1 are correct and each possible outcome of Q-Flip appears at least m_0 times and at most m_1 times then, for any $\lambda > \frac{m_1}{m_0+m_1}$, protocol* `Weak 2-Cast` *fails to achieve weak 2-cast with a probability of at most $P_{r_0} \leq \mathcal{H}(m, m_1, m_0, \lambda m_0)$.*[10]

Proof. The only way for r_0 to make the protocol fail is to make r_1 adopt $y_1 := y_{01} \neq x_s$ during step 5 of the protocol. Hence the following conditions must hold:

- $(y_{01} = 1 - y_1 = 1 - x_s)$,
- $(|\rho_{01}| \geq m_0)$,
- $(|\{i \in \rho_{01} \setminus \sigma_1 : Q_1[i] = 2\}| \geq \lambda\,|\rho_{01}|)$.

Let $u = |\rho_{01}| \geq m_0$. Since s is correct and hence

$$\{i \in \{1,\ldots,m\} : Q_s[i] = 1 - x_s = y_{01} \ \wedge \ Q_1[i] = 2\} \subseteq \sigma_s = \sigma_1 \ ,$$

r_0 must select u indices i such that for λu of them it holds that $i \notin \sigma_s$, and $Q_0[i] = y_{01}$, and $Q_1[i] = 2$. An optimal strategy in order to achieve this is by

[10] See Appendix B for the definition of $\mathcal{H}$.

randomly selecting m_0 indices i such that $Q_0[i] = 1 - x_s$ (corresponding to random selections from the first and the last row in Figure 1–C).
This process corresponds to a hypergeometric distribution with $N = m$, $K = m_1$, and $n = m_0$ (see Section B). The probability for r_0 to succeed is hence given by the tail of this distribution according to $k \geq \lambda m_0$. By Equation (6), for any $\lambda > \frac{m_1}{m_0+m_1}$, this probability can be estimated as

$$P_{r_0} \leq \mathcal{H}(m, m_1, u, \lambda u) \leq \mathcal{H}(m, m_1, m_0, \lambda m_0) \leq e^{-2\left(\lambda - \frac{m_1}{m_0+m_1}\right)^2 m_0} .$$

□

Lemma 9. *For every desired security parameter $k > 0$ there exist parameters m, m_0, and λ such that protocol* `Weak 2-Cast` *has communication and computation complexities polynomial in k and achieves weak 2-cast with an error probability of at most $P_f < e^{-k}$.*

Proof. We let $\lambda_0 = \frac{3}{4}$ and $\lambda_1 = \frac{5}{4}$, and fix the parameterization of the protocol as follows such that there is only one free parameter left, namely m, the number of Q-Flip invocations:

$$\boxed{\begin{array}{lll} m_0 = \lambda_0 \frac{m}{6} & = \frac{m}{8} \\ m_1 = \lambda_1 \frac{m}{6} & = \frac{5m}{24} \\ \lambda \ \ = \lambda_0 & = \frac{3}{4} \end{array}}$$

Now, as a function of security parameter k, let $m \overset{!}{\geq} 288\,(k+2)$. According to Corollary 2 and Lemmas 4, 7, and 8 we get the following estimations:

$$P_{stats} \leq 6 \max\left(\mathcal{C}_{\downarrow}(\frac{1}{6}, m, \lambda_0), \mathcal{C}_{\uparrow}(\frac{1}{6}, m, \lambda_1)\right) \leq 6e^{-\frac{m}{288}} \tag{7}$$

$$P_{r_1} = 0 \tag{8}$$

$$P_s \leq \left(\frac{m_1}{m_0+m_1}\right)^{(1-\lambda)m_0} = \left(\frac{\lambda_1}{\lambda_0+\lambda_1}\right)^{(1-\lambda)\lambda_0 m} = \left(\frac{8}{5}\right)^{-\frac{m}{32}} < e^{-\frac{m}{69}} \tag{9}$$

$$P_{r_0} \leq e^{-2\left(\lambda - \frac{m_1}{m_0+m_1}\right)^2 m_0} = e^{-2\left(\lambda - \frac{\lambda_1}{\lambda_0+\lambda_1}\right)^2 m_0} = e^{-\frac{3m}{128}} \leq e^{-\frac{m}{43}} \tag{10}$$

Finally, the overall error probability can be estimated by the sum of the probabilities that either the statistics of Q-Flip fail, i.e., that at least one of the six possible outcomes appears less than m_0 or more than m_1 times, or that, given good statistics, a faulty player can nevertheless successfully misbehave:

$$P_f \leq P_{stats} + \max(P_{r_1}, P_s, P_{r_0}) \leq 6e^{-\frac{m}{288}} + e^{-\frac{m}{69}} < 7e^{-\frac{m}{288}} < e^{-\frac{m-576}{288}} \leq e^{-k} .$$

□