Two-Threshold Broadcast and Detectable Multi-Party Computation

Matthias Fitzi[1] *, Martin Hirt[2], Thomas Holenstein[2] †, and Jürg Wullschleger[2] †

[1] University of California, Davis, USA, fitzi@cs.ucdavis.edu
[2] ETH Zurich, Switzerland, {hirt,holenst,wjuerg}@inf.ethz.ch

Abstract. Classical distributed protocols like broadcast or multi-party computation provide security as long as the number of malicious players f is bounded by some given threshold t, i.e., $f \leq t$. If f exceeds t then these protocols are completely insecure.

We relax this binary concept to the notion of two-threshold security: Such protocols guarantee full security as long as $f \leq t$ for some small threshold t, and still provide some degraded security when $t < f \leq T$ for a larger threshold T. In particular, we propose the following problems.

○ Broadcast with Extended Validity: Standard broadcast is achieved when $f \leq t$. When $t < f \leq T$, then either broadcast is achieved, or every player learns that there are too many faults. Furthermore, when the sender is honest, then broadcast is always achieved.

○ Broadcast with Extended Consistency: Standard broadcast is achieved when $f \leq t$. When $t < f \leq T$, then either broadcast is achieved, or every player learns that there are too many faults. Furthermore, the players agree on whether or not broadcast is achieved.

○ Detectable Multi-Party Computation: Secure computation is achieved when $f \leq t$. When $t < f \leq T$, then either the computation is secure, or all players detect that there are too many faults and abort.

The above protocols for n players exist if and only if $t = 0$ or $t + 2T < n$.

1 Introduction

1.1 Broadcast

A *broadcast protocol* allows a sender to distribute a value among a set of players such that even a malicious sender cannot make different players receive different values, i.e., a broadcast protocol must satisfy two properties: *validity*, meaning that an honest sender's intended value is received by all players, and *consistency*, meaning that all players receive the same value even when the sender is malicious.

The first broadcast protocols were proposed by Lamport, Shostak, and Pease [LSP82], once for the model with n players connected with bilateral authenticated channels where at most $t < n/3$ players are corrupted, and once for the model with a public-key infrastructure (PKI) and at most $t < n$ corruptions.

* Partially supported by the Packard Foundation.

† Supported by the Swiss National Science Foundation, project no. 2000-066716.01/1.

Both bounds are tight [LSP82,KY84]. The first efficient broadcast protocols were given in [DS82,DFF$^+$82]. Note that a PKI setup can also allow for unconditional security, as shown by Pfitzmann and Waidner [PW96]. More generally, a precomputation phase where broadcast is temporarily achievable can be exploited such that broadcast unconditionally secure against any number of corrupted players is achievable after the precomputation [BPW91,PW96].

1.2 Multi-Party Computation

Secure multi-party computation (MPC) protocols allow a set of n players to securely compute any agreed function on their private inputs, where the following properties must be satisfied: *privacy*, meaning that the corrupted players do not learn any information about the other players' inputs (except for what they can infer from the function output), and *correctness*, meaning that the protocol outputs the correct function value, even when the malicious players misbehave.

The MPC problem was proposed by Yao [Yao82] and first solved by Goldreich, Micali, and Wigderson [GMW87]. This protocol is secure with respect to a computationally bounded adversary corrupting up to $t < n/2$ players, which is optimal. When secure bilateral channels are available, security is achievable with respect to an unbounded adversary that corrupts up to $t < n/3$ players [BGW88,CCD88]; also this bound is tight. For a model assuming broadcast, non-robust protocols for MPC computationally secure against $t < n$ corrupted players are given in [GMW87,BG89a,Gol01].

Broadcast is a key ingredient for MPC protocols, and must be simulated with a respective subprotocol. It fact, the necessary conditions for MPC are due to the requirement of broadcast simulation. When secure broadcast channels are given, then unconditionally secure MPC is achievable even for $t < n/2$ [Bea89,RB89,CDD$^+$99]. Recent results [FGM01,FGMR02] imply that MPC unconditionally secure against $t < n/2$ corruptions is achievable even without assuming broadcast channels in a way that all security conditions are satisfied except for robustness, so called *detectable MPC*.

1.3 Previous Work on Multi-Threshold Security

The first steps towards multi-threshold security were taken by Lamport [Lam83] by analyzing the "weak Byzantine generals" problem, where standard broadcast must be achieved when no player is corrupted ($t = 0$), but agreement among the recipients must be achieved for up to T corruptions. He proved that, deterministically, even this weak form of broadcast is not achievable for $T \geq n/3$.

In [FGM01,FGMR02] a probabilistic protocol called *detectable broadcast* was given that achieves broadcast when no player is corrupted ($t = 0$), but when any minority of the players is corrupted ($T < n/2$) still guarantees that either broadcast is achieved or that all correct players safely abort the protocol. This bound was improved to $T < n$ in [FGH$^+$02].

In another line of research, Vaidya and Pradhan [VP93] proposed "degradable agreement", where broadcast must be achieved when up to t players are corrupted, and some weakened validity and consistency conditions must be achieved when up to T players are corrupted, namely that all players receive either the sent value or $\perp$. However, even when $f \leq t$, the players do not reach agreement on the fact whether or not all players have received the sent value. Degradable agreement is achievable if and only if $2t + T < n$.

1.4 Contributions

We generalize the standard notion of threshold security to two-threshold security where, for two thresholds t and T with $t \leq T$, full security must be achieved when at most $f \leq t$ players are corrupted, and some alleviated form of security must be achieved when $f \leq T$ players are corrupted. This notion is applied to broadcast, resulting in *two-threshold broadcast* with the following two variants:

- BROADCAST WITH EXTENDED VALIDITY: Standard broadcast is achieved when at most $f \leq t$ players are corrupted. When up to $f \leq T$ players are corrupted then still *validity* is guaranteed, i.e., that a correct sender can distribute a value of his own choice among the players.
- BROADCAST WITH EXTENDED CONSISTENCY: Standard broadcast is achieved when at most $f \leq t$ players are corrupted. When up to $f \leq T$ players are corrupted then still *consistency* is guaranteed, i.e., that all players receive the same value, even if the sender is corrupted.

We prove that two-threshold broadcast among n players is achievable if and only if $t = 0$ or $t{+}2T < n$, and construct efficient protocols for all achievable cases (solutions for the special case $t = 0$ were known before [Hol01,FGH$^+$02,GL02]). Moreover, the proposed protocols additionally achieve detection for the case that full broadcast cannot be achieved. The protocol with extended validity additionally achieves that, in case that consistency has not been reached, all players learn this fact (*consistency detection*); and the protocol with extended consistency additionally achieves agreement about the fact whether or not validity has been achieved (*validity detection*).

Finally, we apply the generalized notion to secure multi-party computation (MPC), respectively to detectable precomputation [FGM01,FGMR02]: If up to t players are corrupted then the precomputation succeeds and all correct players accept the precomputation. If up to T players are corrupted then all correct players either commonly accept or commonly reject the precomputation, whereas acceptance implies that the precomputation succeeded. If such a precomputation succeeds then broadcast with full resilience ($t < n$) and secure multi-party computation for $t < n/2$ will be achievable from now on. In other words, with help of detectable precomputation, any protocol in a model with pairwise communication and broadcast can be transformed into a non-robust protocol in the corresponding model *without broadcast.* Detectable precomputation is achievable if and only if $t = 0$ or $t + 2T < n$.

2 Preliminaries

2.1 Models

We consider a set $P = \{p_1, \ldots, p_n\}$ of players, connected by a complete synchronous network of pairwise authenticated (or secure) channels. There is no PKI set up among the players and we assume the presence of an adaptive active adversary. The adversary's computational power is assumed to be unlimited — however, our results are proven tight even with respect to a non-adaptive probabilistic polytime adversary. The model with authenticated channels is denoted by $\mathcal{M}_{\text{aut}}$; the model with secure channels is denoted by $\mathcal{M}_{\text{sec}}$. When referring to their corresponding models from the literature where broadcast channels are additionally given among the players we use the notations $\mathcal{M}_{\text{aut}}^{\text{bc}}$ and $\mathcal{M}_{\text{sec}}^{\text{bc}}$.

2.2 Definitions

A broadcast protocol allows a player (the sender) to consistently send a message to all other players such that all correct players receive the sender's intended value if the sender is correct, but guaranteeing that all correct players receive the same value even when the sender is corrupted.

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

- **Validity:** *If at most t players are corrupted and the sender p_s is correct then all correct players p_i decide on the sender's input value, $y_i = x_s$.*
- **Consistency (or Agreement):** *If at most t players are corrupted then all correct players decide on the same output value, i.e., if $p_i, p_j \in P$ are correct then $y_i = y_j$.* ◇

In this paper, we focus on binary broadcast (domain $\mathcal{D} = \{0, 1\}$) since broadcast for any finite domain $\mathcal{D}$ can be efficiently reduced to the binary case [TC84].

Our first generalization of standard broadcast demands validity even when the number f of corrupted players exceeds t, called *broadcast with extended validity*. We directly give a strengthened definition that allows the players to learn whether or not consistency has been achieved. For this we have the players p_i decide on an additional binary grade value g_i, $g_i = 1$ implying (but not being equivalent with) the fact that consistency has been achieved.[3] It can be guaranteed that consistency is always detected if $f \leq t$ ("completeness") but never incorrectly detected if $f \leq T$ ("soundness").

[3] Note that, for the interesting case $T \geq n/3$, it is not possible to achieve that $g_i = 1$ is equivalent with having achieved consistency since this would immediately imply standard broadcast for $T \geq n/3$.

Definition 2 (ExtValBC). *A protocol Ψ among P where player $p_s \in P$ (called* the sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ and a grade value $g_i \in \{0, 1\}$ achieves* broadcast with extended validity and consistency detection (ExtValBC) *with respect to thresholds t and T ($T \geq t$) if it satisfies the following conditions:*

- **Broadcast:** *If at most $f \leq t$ players are corrupted then every correct player p_i decides on the same pair of outputs $(y, 1)$, i.e., $y_i = y$ and $g_i = 1$. Furthermore, if the sender p_s is correct then $y_i = x_s$.*
- **Extended Validity:** *If $f \leq T$ and the sender p_s is correct then every correct player p_i decides on the sender's input value, $y_i = x_s$.*
- **Consistency Detection:** *If $f \leq T$ and any correct player p_i computes $g_i = 1$ then every correct player p_j computes $y_j = y_i$.* ◇

Our second generalization of standard broadcast demands consistency even when the number f of corrupted players exceeds t, called *broadcast with extended consistency*. Again, we directly give a strengthened definition that allows the players to learn whether or not validity has been achieved. In contrast to the inherently non-common consistency detection in ExtValBC for $T \geq n/3$, here we require that the players decide on the same grade output g_i. If $f \leq t$ then validity is always detected ("completeness"), and if $f \leq T$ then the detection of validity always implies validity ("soundness").

Definition 3 (ExtConsBC aka Detectable Broadcast). *A protocol Ψ among P where player $p_s \in P$ (called* the sender*) holds an input value $x_s \in \mathcal{D}$ and every player $p_i \in P$ finally decides on an output value $y_i \in \mathcal{D}$ and a grade value $g_i \in \{0, 1\}$ achieves* broadcast with extended consistency and validity detection (ExtConsBC) *with respect to thresholds t and T ($T \geq t$) if it satisfies the following conditions:*

- **Broadcast:** *If at most $f \leq t$ players are corrupted then every correct player p_i decides on the same pair of outputs $(y, 1)$, $y_i = y$ and $g_i = 1$. Furthermore, if the sender p_s is correct then $y_i = x_s$.*
- **Extended Consistency:** *If $f \leq T$ then every correct player p_i decides on the same pair of outputs (y, g), $y_i = y$ and $g_i = g$.*
- **Validity Detection:** *If $f \leq T$, the sender p_s is correct, and any correct player p_i computes $g_i = 1$, then $y_i = x_s$.* ◇

Along the lines of [BPW91,PW96] detectable broadcast can be turned into a "detectable precomputation" for future broadcast unconditionally secure against any number of corrupted players, $t < n$.

Definition 4 (Detectable Precomputation). *A protocol among n players where every player $p_i \in P$ computes some private data Δ_i and finally decides on a decision bit $g_i \in \{0, 1\}$ achieves* detectable precomputation for broadcast *(or* detectable precomputation*, for short) with respect to thresholds t and T ($T \geq t$) if it satisfies:*

- **Validity (or Robustness):** *If at most $f \leq t$ players are corrupted then the correct players accept ($g_i = 1$).*
- **Consistency (or Correctness):** *If $f \leq T$ then all correct players commonly accept ($g_i = 1$) or commonly reject ($g_i = 0$) the protocol; moreover, if the private data Δ_i held by all correct players is inconsistent in the sense that it does not guarantee for arbitrarily resilient broadcast then the correct players reject ($g_i = 0$).*
- **Independence (or Fairness):** *At the time of the precomputation, a correct player does not yet need to know the value to be broadcast later.* ◇

2.3 Protocol Notation

Protocols are specified with respect to player set P (where $|P| = n$) and stated with respect to the local view of player p_i, meaning that all players $p_i \in P$ execute this code in parallel with respect to their own identity i. Player p_i's input is called x_i. Player p_i's output value is written as y_i, or g_i, or pair (y_i, g_i) and will always be obvious from the context. For simplicity, it is not explicitly stated how to handle values received from corrupted players that are outside the specified domain. Such a value is always implicitly assumed to be replaced by a default value inside the specified domain.

3 Broadcast with Extended Validity

In this section, we present an efficient solution for broadcast with extended validity and consistency detection, ExtValBC. The model is $\mathcal{M}_{\mathrm{aut}}$ and the given protocol achieves perfect security. Since, for the special case $t = 0$, efficient and optimally resilient protocols were already given in [Hol01,FGH$^+$02], we focus on protocols for $t > 0$.

The construction in this section works along the lines of the phase-king paradigm of [BG89b,BGP89]. An important building block for phase-king protocols is graded consensus, a derivative of graded broadcast [FM97].

Definition 5 (Graded Consensus GC). *A protocol among P where every player $p_i \in P$ holds an input value $x_i \in \mathcal{D}$ and finally decides on an output value $y_i \in \mathcal{D}$ and a grade $g_i \in \{0, 1\}$ achieves* graded consensus *with respect to threshold t if it satisfies the following conditions:*

- **Validity (or Persistency):** *If at most t players are corrupted and all correct players p_i hold the same input value $x_i = v$ then all correct players p_i decide on it, $y_i = v$, and get grade $g_i = 1$.*
- **Consistency Detection:** *If at most t players are corrupted and any correct player p_i gets grade $g_i = 1$ then all correct players p_j decide on the same output value, $y_i = y_j$.* ◇

We now generalize graded consensus to ExtValGC in the same way as broadcast was generalized to ExtValBC in Section 2.2. Since graded consensus already involves a grade g_i for consistency detection, we do not add an additional grade value for ExtValGC but simply extend the grade range to $g_i \in \{0,1,2\}$ whereas $g_i = 1$ implies consistency detection if at most $f \leq t$ players are corrupted and $g_i = 2$ implies consistency detection if at most $f \leq T$ players are corrupted.

Definition 6 (ExtValGC). *A protocol among P where each player $p_i \in P$ holds an input value x_i and finally decides on an output value y_i and a grade value $g_i \in \{0,1,2\}$ achieves* graded consensus with extended validity and consistency detection (ExtValGC) *with respect to thresholds t and T ($T \geq t$) if it satisfies the following conditions:*

Validity: *If at most $f \leq T$ players are corrupted and every correct player p_i enters the protocol with the same input value $x_i = v$ then every correct player p_i computes outputs $y_i = v$ and $g_i \geq 1$, and in particular, $g_i = 2$ if at most t players are corrupted.*

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

Protocol 1 `ExtValGC`(P, x_i, t, T)

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

Lemma 1 ("Two-threshold weak consensus"). *In model $\mathcal{M}_{\text{aut}}$, if $t + 2T < n$, Protocol 1 satisfies the following properties.*

Validity: *If at most $f \leq T$ players are corrupted and every correct player p_i holds the same input value $x_i = v$ then every correct player holds value $z_i = x_i = v$ after step 3 of the protocol.*

Consistency: *If $f \leq t$ and any correct player p_i holds value $z_i \in \{0,1\}$ after step 3 then every correct player p_j holds value $z_j \in \{z_i, \perp\}$ after step 3.*

Proof. If at most $f \leq T$ players are corrupted and every correct player p_i holds the same input value $x_i = v$ then, for every correct player p_i, it holds that $|S_i^v| \geq n - T$ and hence every such p_i computes $z_i = x_i = v$.

If $f \le t$ and any correct player p_i holds value $z_i \in \{0,1\}$ after step 3 then $|S_i^{z_i}| \ge n-T$ and thus, for every correct player p_j, it holds that $|S_j^{z_i}| \ge n-T-t > T$ and thus $z_j \in \{z_i, \perp\}$ after step 3. □

Lemma 2 (ExtValGC). *In model $\mathcal{M}_{\mathrm{aut}}$, if $t+2T < n$ (and $T \ge t$), Protocol 1 achieves perfectly secure ExtValGC with respect to thresholds t and T.*

Proof.
Validity: Suppose that $f \le T$, and that every correct player p_i enters the protocol with the same input value $x_i = v$. Then, by Lemma 1 (validity), every correct player p_i holds value $z_i = v$ at the end of step 3 and thus value v is redistributed by all correct players in step 4. Thus $|U_i^v| \ge n-T > T$, and every correct player computes $y_i = v$ and $g_i \ge 1$. Furthermore, if only $f \le t$ players are corrupted then $|U_i^v| \ge n-t$, and every correct player p_i computes $g_i = 2$.
Consistency Detection: Suppose that $f \le t$, and that some correct player p_i computes $g_i \ge 1$ and $y_i = v \in \{0,1\}$.

Let $\mathcal{C}$ be the set of corrupted players, $\mathcal{S}^v$ be the set of correct players who sent value v in step 1, and let $\mathcal{U}^v$ be the set of correct players who sent value v in step 4. Note that $\mathcal{S}^v = S_j^v \backslash \mathcal{C}$ and $\mathcal{U}^v = U_j^v \backslash \mathcal{C}$ for any j.

Since $g_i \ge 1$ we have that $|U_i^v| \ge n-T$ and thus that $|\mathcal{U}^v| \ge n-T-t$. Since a correct player p_j can only change to $z_j :=\perp$, it follows that $|\mathcal{U}^v| \le |\mathcal{S}^v|$. Therefore, for every player p_j, $|S_j^v| \ge |\mathcal{S}^v| \ge n-T-t$. The bound $n > 2T+t$ now implies that $|S_j^{1-v}| \le T+t < n-T$ and therefore that $\mathcal{U}^{1-v} = \emptyset$. Thus, $|U_j^{1-v}| \le |\mathcal{C}| \le t$ and $|U_j^v| \ge |\mathcal{S}^v| \ge n-T-t > T$, and $y_j = y_i$.

Assuming that at most $f \le T$ players are corrupted and that $g_i = 2$, it follows that $|\mathcal{U}^v| \ge n-t-T$. This implies that $\mathcal{U}^{1-v} = \emptyset$ and that, again, $|U_j^v| > |U_j^{1-v}|$. □

The final protocol for ExtValBC can now be built from ExtValGC according to the phase-king paradigm [BG89b,BGP89]. The only difference to the standard phase-king structure is an additional round of ExtValGC at the end of the protocol in order to allow for consistency detection.

Protocol 2 $\mathtt{ExtValBC}_{p_1}(P, x_1, t, T)$

```
1. if $i = 1$ then SendToAll($x_1$) fi;    $P$: Receive($y_i$);
2. for $k := 2$ to $t+1$ do
3.     $(y_i, h_i) :=$ ExtValGC$(P, y_i, t, T)$;
4.     if $i = k$ then SendToAll($y_i$) fi;    $P$: Receive($y_i^k$);
5.     if $h_i = 0$ then $y_i := y_i^k$ fi;
6. od;
7. $(y_i, h_i) :=$ ExtValGC$(P, y_i, t, T)$;
8. if $h_i = 2$ then $g_i := 1$ else $g_i := 0$ fi;
9. return $(y_i, g_i)$;
```

Theorem 1 (ExtValBC). *In model $\mathcal{M}_{\mathrm{aut}}$, if $t + 2T < n$ (and $T \ge t$), Protocol 2 efficiently achieves perfectly secure ExtValBC (with sender p_1) with respect to thresholds t and T.*

Proof. To prove that the conditions for broadcast and extended validity hold, we show that validity holds for $f \leq T$, and consistency for $f \leq t$.
VALIDITY: Suppose that at most $f \leq T$ players are corrupted and that the sender p_1 is correct. Then, by the validity property of ExtValGC, every correct player p_i finally computes $y_i = x_1$ at the end of the protocol.
CONSISTENCY: If $f \leq t$ then there is a player $p_\ell \in \{p_1, \ldots, p_{t+1}\}$ that is correct. At the end of phase $k = \ell$, every correct player p_i holds the same value $y_i = y_\ell = v$ which, by the validity property of ExtValGC, stays persistent until step 7 of the protocol and every correct player finally computes $y_i = v$, $h_i = 2$, and thus $g_i = 1$.
CONSISTENCY DETECTION: Assume that $f \leq T$ and some correct player p_i computes $g_i = 1$ at the end of the protocol. This implies that $h_i = 2$ after the invocation of ExtValGC in step 7, and by the consistency-detection property of ExtValGC, that every correct player p_j computed $y_j = y_i$ during this invocation and thus terminated the protocol with $y_j = y_i$. □

Theorem 2 (Impossibility of ExtValBC). *In standard models $\mathcal{M}_{\text{sec}}$ and $\mathcal{M}_{\text{aut}}$, ExtValBC among a set of n players $P = \{p_0, \ldots, p_{n-1}\}$ is impossible if $t > 0$ and $t + 2T \geq n$. For every protocol there exists a value $x_0 \in \{0, 1\}$ such that, when the sender holds input x_0, the adversary can make the protocol fail with a probability of at least $1/6$ if it is computationally bounded, and with a probability of at least $1/3$ if it is computationally unbounded.*

Proof. Assume Ψ to be a protocol for ExtValBC among n players $p_0, \ldots, p_{n-1}$ with sender p_0 that tolerates $t > 0$ and $t + 2T \geq n$.

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

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

Along the lines of [FLM86], instead of connecting the original processors as required for the broadcast setting, we build a network involving all $2n$ processors (i.e., the original ones together with their copies) by arranging the six processor sets Π_k in a circle. In particular, for all sets Π_k $(0 \leq k \leq 5)$, every processor $\pi_i \in \Pi_k$ is connected (exactly) by one channel with all processors in $\Pi_k \backslash \{\pi_i\}$, $\Pi_{(k-1) \bmod 6}$, and $\Pi_{(k+1) \bmod 6}$. Hence, each processor π_i in the new system is symmetrically connected with exactly one processor of each type (different from his own one) as in the original system. We say that Π_k and Π_ℓ are *adjacent processor sets* if and only if $\ell \equiv k \pm 1 \pmod 6$.

Now, for every set $\Pi_k \cup \Pi_{(k+1) \bmod 6}$ $(0 \leq k \leq 5)$ in the new system and without the presence of an adversary, their common view is indistinguishable

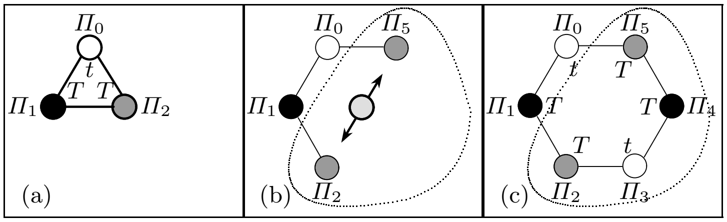

Fig. 1. *Rearrangement of processors in the proof of Theorem 2*

from their view as the set of processors $\Pi_{k \bmod 3} \cup \Pi_{(k+1) \bmod 3}$ in the original system with respect to an adversary who corrupts all (up to either t or T) processors of the remaining processor set $\Pi_{(k+2) \bmod 3}$ in an admissible way.

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

By the extended-validity property of ExtValBC, the at least $n-T$ processors $p_i \in \Pi_0 \cup \Pi_1$ must compute $y_i = x_0$, the processors $p_i \in \Pi_0 \cup \Pi_5$ must compute $y_i = x_0$, and the processors $p_i \in \Pi_3 \cup \Pi_2$ and $p_i \in \Pi_3 \cup \Pi_4$ must compute $y_i = x_n = 1 - x_0$. By the broadcast property of ExtValBC, the at least $n - t$ processors $p_i \in \Pi_1 \cup \Pi_2$ must compute the same value $y_i = v$ and the processors $p_i \in \Pi_4 \cup \Pi_5$ the same value $y_i = w$.

Hence, for any possible run of the new system on inputs x_0 and $x_n = 1 - x_0$, chosen a pair $(\Pi_k, \Pi_{(k+1) \bmod 6})$ of processor sets uniformly at random, the probability that the conditions for ExtValBC are violated for this pair is at least $1/3$. In particular, there is a pair $(\Pi_k, \Pi_{(k+1) \bmod 6})$ in the new system such that, over all possible runs on inputs $x_0 = 0$ and $x_n = 1$ the probability that the conditions for ExtValBC are violated for $(\Pi_k, \Pi_{(k+1) \bmod 6})$ is at least $1/3$.

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

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

with a probability of at least $1/3$. If the adversary is computationally bounded then it can still make the protocol fail with a probability of at least $1/6$. □

4 Broadcast with Extended Consistency

We directly present an efficient solution for detectable precomputation which is strictly stronger than broadcast with extended consistency. Since, for the special case $t = 0$, efficient and optimally resilient protocols were already given in [FGH$^+$02], we focus on protocols for $t > 0$. In order to achieve unconditional security, model $\mathcal{M}_{\text{sec}}$ is required.[4]

Recall that the Pfitzmann-Waidner protocol [PW96], in model $\mathcal{M}_{\text{sec}}^{\text{bc}}$, efficiently precomputes for future broadcast in plain model $\mathcal{M}_{\text{sec}}$ (without broadcast) unconditionally secure against any number of corrupted players. Our protocol for detectable precomputation basically consists of an instance of this protocol (designed for model $\mathcal{M}_{\text{sec}}^{\text{bc}}$) wherein each invocation of a broadcast channel is replaced by an invocation of ExtValBC (designed for model $\mathcal{M}_{\text{aut}}$).

Protocol 3 `DetPrecomp`(P)

1. *Execute the Pfitzmann-Waidner protocol for $b + n$ future broadcasts wherein each invocation of broadcast is replaced by ExtValBC Protocol 2 with respect to thresholds t and T. Of these instances, b are computed with respect to the intended senders $s \in \{1, \ldots, n\}$ of the future broadcasts. Of the other n instances, one is computed with respect to each player $p_i \in P$ as a future sender.*
2. *Every player p_i computes $\Gamma_i := G_i = \bigwedge_k g_i^k$ where the g_i^k are all grades received during an invocation of ExtValBC in step 1.*

   Synchronize: Wait and start executing the next step at round $\lfloor \frac{n^2(9t+10)}{2} \rfloor + 1$.
3. *Send value G_i to each other player; receive the values $G_i^1, \ldots, G_i^n$.*
4. *For every player $p_j \in P$ an instance of Pfitzmann-Waidner broadcast with resilience T is invoked based on the (not necessarily consistent) PKI consisting of the information exchanged during step 1 of the protocol — where p_j inputs Γ_j as a sender.[5] Store the received values Γ_i^j $(j \in \{1, \ldots, n\})$.*
5. *Compute $g_i = 1$ if $|\{j \mid G_i^j = 1\}| > T \ \wedge \ |\{j \mid \Gamma_i^j = 1\}| \geq n - t$ and $g_i = 0$, otherwise.*

Theorem 3 (Detectable precomputation). *In model $\mathcal{M}_{\text{sec}}$, for any integer $b > 0$ and security parameter $\kappa > 0$, if $t + 2T < n$ (and $T \geq t$), Protocol 3 achieves unconditionally secure detectable precomputation for b later broadcasts among n players with respect to thresholds t and T with the following properties:*

Protocol 3 has computation and communication complexities polynomial in n, b, and linear in κ. The error probability of any future broadcast is $\varepsilon < 2^{-\kappa}$. The correct players all terminate the protocol during the same communication round. Furthermore, the computation and communication complexities can be reduced to polynomial in n, $\log b$, and linear in κ.

Proof.

Validity: Suppose that $f \leq t$ players are corrupted. Hence, according to the definition of ExtValBC, all invocations of Protocol 2 achieve broadcast (when neglecting the grade outputs) and that every correct player p_i computes $g_i = 1$. Thus, the players share a consistent PKI, all correct players p_i compute $G_i = 1$,

[4] For the case of computational security, there is a simpler solution for model $\mathcal{M}_{\text{aut}}$.

[5] Note that such an instance does not necessarily achieve broadcast. However, even then, it will always efficiently terminate after $T + 1$ rounds as can be seen by examination [DS82].

and all broadcast invocations in step 4 indeed achieve broadcast. Thus, in steps 3 and 4, the players p_i compute values G_i^j and Γ_i^j such that $|\{j \mid G_i^j = 1\}| \geq n-t > T$ and $|\{j \mid \Gamma_i^j = 1\}| \geq n - t$. Finally, all correct players p_i compute $g_i = 1$.

Consistency: Suppose $f \leq T$. If every correct player p_i rejects by computing $g_i = 0$ then consistency is satisfied. Thus, suppose that some correct player p_i accepts by computing $g_i = 1$, implying that some correct player p_k computed $G_k = 1$. Thus, according to the definition of ExtValBC, all invocations of Protocol 2 achieved broadcast (when neglecting the grade outputs), and the players share a consistent PKI. Hence, all broadcast invocations in step 4 indeed achieve broadcast and all correct players p_j compute the same set of values $\Gamma_j^1, \ldots, \Gamma_j^n$. Since $g_i = 1$, for every correct player p_ℓ it holds that $|\{j \mid \Gamma_\ell^j = 1\}| \geq n - t$ and thus that $|\{j \mid G_\ell^j = 1\}| \geq n - t - T > T$, and all players p_ℓ compute $g_\ell = 1$.

Independence, error probability, and complexities: Independence follows from the structure of the Pfitzmann-Waidner protocol.

Executing the Pfitzmann-Waidner protocol with security parameter κ guarantees each single of the $b + n$ broadcasts to have an error probability of $\varepsilon < 2^{-\kappa}$ [PW96]. The error probability of each of the b "net" broadcasts is given by the probability that one of the n broadcasts during step 4 fails and the probability that the one broadcast fails given that those n broadcasts reliably worked, which is bounded by $(n + 1)$ times the error probability of one single broadcast precomputed for with the Pfitzmann-Waidner protocol. Executing the Pfitzmann-Waidner protocol with security parameter $\kappa_0 \geq \kappa + \lceil \log(n + b) \rceil$ hence bounds the error probability of any single "net" broadcast to $\varepsilon < 2^{-\kappa}$.

Efficiency follows from [PW96] and the construction of Protocol 3. That all players terminate the protocol during the same communication round is ensured by the synchronization procedure at the end of step 2: in the Pfitzmann-Waidner protocol, the worst-case number of rounds for any player is at most $\lfloor \frac{n^2(9t+10)}{2} \rfloor$ [PW96]. Finally, in order to get polylogarithmic dependence on b, the regeneration techniques in [PW96] can be applied. □

The above construction for detectable precomputation immediately allows for broadcast with extended consistency and validity detection:

Theorem 4 (ExtConsBC). *In model $\mathcal{M}_{\text{sec}}$ efficient unconditionally secure ExtConsBC with respect to thresholds t and T is possible if $t + 2T < n$ (and $T \geq t$).*

Proof. In order to achieve ExtConsBC, the players first execute a detectable precomputation with Protocol 3. If the precomputation fails (there are more than t corrupted players), then every player p_i sets his output value to $y_i =\perp$ and his grade to $g_i = 0$. If the precomputation succeeds (which is guaranteed if at most t players are corrupted), then a valid setup for further Pfitzmann-Waidner broadcast is established. Then the players invoke an instance of Pfitzmann-Waidner broadcast (using this setup), which tolerates any number of corrupted players, and set $g_i = 1$. □

Theorem 5 (Impossibility of ExtConsBC). *In standard models $\mathcal{M}_{\text{sec}}$ and $\mathcal{M}_{\text{aut}}$, ExtConsBC among a set of n players $P = \{p_1, \ldots, p_n\}$ is impossible if $t > 0$ and $t + 2T \geq n$. For every protocol there exists a value $x_s \in \{0, 1\}$ such that, when the sender p_s holds input x_s, the adversary can make the protocol fail with a probability of at least $1/6$ if it is computationally bounded, and with a probability of at least $1/3$ if it is computationally unbounded.*

Proof. Note that a direct proof similar to the one for Theorem 2 would be possible. However, here we use a reduction argument: For the sake of contradiction, assume that ExtConsBC is possible with respect to thresholds t and T such that $t > 0$ and $t + 2T \geq n$ (and error probability below $1/6$, respectively $1/3$). Such a protocol can be transformed into a broadcast protocol with extended validity (with the same error probabilities) as follows: If ExtConsBC with sender p_s succeeds (which can be consistently detected by all players) with player p_i receiving y_i, then p_i outputs y_i and terminates. If ExtConsBC fails (there are more than t corrupted players), then the sender p_s sends his input x_s to all players, and every player outputs the received value. Obviously, this protocol achieves broadcast for up to t corrupted players and extended validity with consistency detection for up to T corrupted players. According to Theorem 2, such a protocol cannot exist with respect to the stated thresholds t and T. Hence ExtConsBC is not achievable with respect to these thresholds. □

5 Detectable Multi-Party Computation

Detectable precomputation immediately allows to turn any protocol Ψ (e.g., a protocol for multi-party computation) in model $\mathcal{M}_{\text{aut}}^{\text{bc}}$ (or $\mathcal{M}_{\text{sec}}^{\text{bc}}$) into a "detectable version" for standard model $\mathcal{M}_{\text{aut}}$ (or $\mathcal{M}_{\text{sec}}$) without broadcast channels. For the case that $f \leq t$ players are corrupted this transformation preserves any security properties of Ψ except for zero-error. For the case that $f \leq T$ players are corrupted the transformation still preserves any security properties of Ψ except for zero-error and robustness. Robustness is lost since detectable precomputation cannot guarantee validity for T (at least for the interesting cases where $T \geq n/3$). Zero-error is lost since there is no deterministic protocol for detectable precomputation as follows from Lamport's result [Lam83].

In particular, it is possible to define the "detectable" version of multi-party computation along the lines of [FGMR02].

Definition 7 (Detectable precomputation for MPC). *Let Ψ be an MPC protocol among P in a model assuming broadcast, model $\mathcal{M}_{*}^{\text{bc}} \in \{\mathcal{M}_{\text{aut}}^{\text{bc}}, \mathcal{M}_{\text{sec}}^{\text{bc}}\}$, and let $\mathcal{M}_{*} \in \{\mathcal{M}_{\text{aut}}, \mathcal{M}_{\text{sec}}\}$ be the same model as $\mathcal{M}_{*}^{\text{bc}}$ but without broadcast. A protocol among P where every player $p_i \in P$ computes some private data Δ_i and finally decides on a decision bit $g_i \in \{0, 1\}$ achieves* detectable precomputation for MPC with Ψ *with respect to thresholds t and T ($T \geq t$), and t', if it satisfies the following conditions:*

Robustness: *If at most $f \leq t$ players are corrupted then the correct players accept ($g_i = 1$).*

Correctness: *If $f \leq T$ then all correct players commonly accept ($g_i = 1$) or commonly reject ($g_i = 0$) the protocol; moreover, if the private data Δ_i held by all correct players is inconsistent in the sense that it does not guarantee for MPC secure against t' corrupted players in model $\mathcal{M}_*$ then the correct players reject ($g_i = 0$).*

Independence: *At the time of the precomputation, a correct player does not yet need to know his input values for the later multi-party computations.* ◇

Together with [Bea89,RB89,CDD$^+$99], detectable precomputation for broadcast trivially allows for detectable MPC such that only robustness is lost since non-zero error is necessary for multi-party computation secure against $t \geq n/3$ corrupted players [DDWY93]. The following theorem follows immediately from Theorem 3:

Theorem 6 (Detectable precomputation for MPC). *Let Ψ be the MPC protocol in [CDD$^+$99] for model $\mathcal{M}_{\mathrm{sec}}^{\mathrm{bc}}$ unconditionally secure against a faulty minority of corrupted players. In model $\mathcal{M}_{\mathrm{sec}}$, detectable precomputation for unconditionally secure MPC with Ψ among n players with respect to thresholds t and T ($T \geq t$), and t', is efficiently achievable if $(t + 2T < n \quad \vee \quad t = 0)$ and $t' < n/2$.*

For the case that $t > 0$ and $t + 2T \geq n$, and $t' \geq n/3$, detectable precomputation for MPC is not even achievable with respect to computational security.

Proof. Achievability follows from [Bea89,RB89,CDD$^+$99] and Theorem 3. Impossibility follows from Theorem 5 together with the impossibility of broadcast for $t \geq n/3$ if no consistent PKI is given. □

Alternatively to this theorem, there are protocols for *non-robust* MPC *without fairness* for model $\mathcal{M}_{\mathrm{aut}}^{\mathrm{bc}}$ [GMW87,BG89a,Gol01] computationally secure against any number of corrupted players. These protocols can be detectably precomputed with help of Protocol 3 (or its more efficient computational analogue), which directly leads to corresponding protocols for the weaker model $\mathcal{M}_{\mathrm{aut}}$ without broadcast (and without need for a PKI setup).

6 General Adversaries

In contrast to threshold adversaries, general adversaries are characterized by the possible subsets of players which might be corrupted at the same time. More precisely, a general adversary is characterized by a collection $\mathcal{Z}$ of subsets of the player set P, i.e., $\mathcal{Z} \subseteq 2^P$. A $\mathcal{Z}$-adversary can corrupt the players of one of the sets in $\mathcal{Z}$. It is known that broadcast secure against a $\mathcal{Z}$-adversary is achievable if and only if no three sets in $\mathcal{Z}$ add up to P [HM97,FM98]. Under the same condition, secure multi-party computation is possible [HM97]. When no PKI is set up these bounds are tight.

Our results on two-threshold security can immediately be generalized to general adversaries. We consider two adversary structures $\mathcal{Z}$ and $\mathcal{Z}^*$, where $\mathcal{Z} \subseteq \mathcal{Z}^*$, corresponding to the threshold case with t and T, where $t \leq T$. A protocol is $(\mathcal{Z}, \mathcal{Z}^*)$-secure if it provides full security against an adversary corrupting a set $Z \in \mathcal{Z}$, and degraded security against an adversary corrupting a set $Z \in \mathcal{Z}^*$. A broadcast protocol with extended validity (consistency) achieves normal broadcast when any set $Z \in \mathcal{Z}$ is corrupted, and still provides validity (consistency) when a set $Z \in \mathcal{Z}^*$ is corrupted. $(\mathcal{Z}, \mathcal{Z}^*)$-secure broadcast is achievable if and only if

$$\forall Z_1 \in \mathcal{Z},\ Z_2 \in \mathcal{Z}^*,\ Z_3 \in \mathcal{Z}^* : Z_1 \cup Z_2 \cup Z_3 \neq P.$$

Given the constructions for two-threshold broadcast in this paper, the construction of such a protocol with respect to general adversaries is straight-forward. The above results immediately generalize to detectable multi-party computation.

7 Conclusions

We generalized the standard notion of broadcast to *two-threshold broadcast*, requiring standard broadcast for the case that $f \leq t$ players are corrupted and either validity or consistency when $t \leq f \leq T$. We showed that, for both cases, (efficient) unconditionally secure two-threshold broadcast is achievable among n players if and only if $t = 0$ or $t + 2T < n$. Our protocol with extended validity additionally achieves that, when consistency is not reached, all players agree on this fact (*consistency detection*); our protocol with extended consistency additionally achieves agreement about the fact whether or not validity is achieved (*validity detection*).

In the same way, detectable precomputation can be generalized with respect to two thresholds t and T. In a model with pairwise channels but without broadcast (and no PKI among the players), such a protocol achieves the following:

- if $f \leq T$ players are corrupted then either all correct players accept or they all reject the protocol outcome. If the correct players accept the protocol outcome then broadcast secure against $t' < n$ corrupted players and MPC secure against $t' < n/2$ corrupted players are achievable from now on.
- if $f \leq t$ then all correct players accept the protocol outcome.

Detectable precomputation is (efficiently) achievable if and only if $t + 2T < n$ or $t = 0$.

8 Acknowledgments

We thank the anonymous referees for their helpful comments.

References

[Bea89] D. Beaver. Multiparty protocols tolerating half faulty processors. In *CRYPTO '89*, vol. 435 of *LNCS*, pp. 560–572. Springer-Verlag, 1989.

[BG89a] D. Beaver and S. Goldwasser. Multiparty computation with faulty majority. In *Proc. 30th FOCS*, pp. 468–473. IEEE 1989.

[BG89b] P. Berman and J. Garay. Asymptotically optimal distributed consensus. In *Proc. 16th International Colloquium on Automata, Languages and Programming*, vol. 372 of *LNCS*, pp. 80–94. Springer-Verlag, 1989.

[BGP89] P. Berman, J. A. Garay, and K. J. Perry. Towards optimal distributed consensus. In *Proc. 30th FOCS*, pp. 410–415. IEEE, 1989.

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

[BPW91] B. Baum-Waidner, B. Pfitzmann, and M. Waidner. Unconditional Byzantine agreement with good majority. In *Proc. 8th Theoretical Aspects of Computer Science*, vol. 480 of *LNCS*, pp. 285–295. Springer-Verlag, 1991.

[CCD88] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proc. 20th STOC*, pp. 11–19. ACM, 1988.

[CDD$^+$99] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations secure against an adaptive adversary. In *EUROCRYPT '99*, vol. 1592 of *LNCS*, pp. 311–326. Springer-Verlag, 1999.

[DDWY93] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of the ACM*, 40(1):17–47, 1993.

[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, 1982.

[DS82] D. Dolev and H. R. Strong. Polynomial algorithms for multiple processor agreement. In *Proc. 14th STOC*, pp. 401–407. ACM, 1982.

[FGH$^+$02] M. Fitzi, D. Gottesman, M. Hirt, T. Holenstein, and A. Smith. Byzantine agreement secure against faulty majorities from scratch. In *Proc. 21st PODC*, ACM, 2002.

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

[FGMR02] M. Fitzi, N. Gisin, U. Maurer, and O. von Rotz. Unconditional Byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In *EUROCRYPT 2002*, vol. 2332 of *LNCS*. Springer-Verlag, 2002.

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

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

[FM98] Matthias Fitzi and Ueli Maurer. Efficient Byzantine agreement secure against general adversaries. In *Proc. 12th DISC*, vol. 1499 of *LNCS*, pp. 134–148. Springer-Verlag, 1998.

[GL02] S. Goldwasser and Y. Lindell. Secure computation without agreement. In *Proc. 16th DISC'02*, vol. 2508 of *LNCS*, pp. 17–32. Springer-Verlag, 2002.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. 19th STOC*, pp. 218–229, ACM, 1987.

[Gol01] O. Goldreich. Secure multi-party computation, working draft, version 1.3, June 2001.

[HM97] Martin Hirt and Ueli Maurer. Complete characterization of adversaries tolerable in secure multi-party computation. In *Proc. 16th PODC*, pp. 25–34. ACM 1997. Full version in *Journal of Cryptology*, 13(1):31–60, 2000.

[Hol01] T. Holenstein. Hybrid broadcast protocols. Master's Thesis, ETH Zurich, October 2001.

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

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

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

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

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

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

[VP93] N. H. Vaidya and D. K. Pradhan. Degradable agreement in the presence of Byzantine faults. In *Proc. 13th International Conference on Distributed Computing Systems*, pp. 237–245. IEEE, 1993.

[Yao82] A. C. Yao. Protocols for secure computations. In *Proc. 23rd FOCS*, pp. 160–164. IEEE, 1982.