Revisiting Fairness in MPC: Polynomial Number of Parties and General Adversarial Structures

Dana Dachman-Soled(✉)

University of Maryland, College Park, USA
danadach@umd.edu

Abstract. We investigate fairness in secure multiparty computation when the number of parties $n = \text{poly}(\lambda)$ grows polynomially in the security parameter, λ. Prior to this work, efficient protocols achieving fairness with no honest majority and polynomial number of parties were known only for the AND and OR functionalities (Gordon and Katz, TCC'09). We show the following:

- We first consider symmetric Boolean functions $F : \{0,1\}^n \to \{0,1\}$, where the underlying function $f_{n/2,n/2} : \{0,\ldots,n/2\} \times \{0,\ldots,n/2\} \to \{0,1\}$ can be computed fairly and efficiently in the 2-party setting. We present an efficient protocol for any such F tolerating $n/2$ or fewer corruptions, for $n = \text{poly}(\lambda)$ number of parties.
- We present an efficient protocol for n-party majority tolerating $n/2 + 1$ or fewer corruptions, for $n = \text{poly}(\lambda)$ number of parties. The construction extends to $n/2 + c$ or fewer corruptions, for constant c.
- We extend both of the above results to more general types of adversarial structures and present instantiations of non-threshold adversarial structures of these types. These instantiations are obtained via constructions of *projective planes* and *combinatorial designs*.

1 Introduction

In secure multiparty computation (MPC), parties compute the joint function of their inputs in a distributed fashion, while keeping their inputs private. Formally defining the security model for MPC is quite complex and there are various different flavors of security such as *computational* vs. *information-theoretic*, *security-with-abort* vs. *fairness* vs. *guaranteed output delivery*, *broadcast-channel* vs. *no broadcast channel*, *rushing* vs. *non-rushing*.

In this work, we focus on the setting of *computationally-secure*, n-party MPC in the presence of a *broadcast channel* with a *rushing* adversary. Further, we will require the *fairness* guarantee, which, informally, states that if one party

Supported in part by NSF grants #CNS-1933033, #CNS-1453045 (CAREER) and by financial assistance awards 70NANB15H328 and 70NANB19H126 from the U.S. Department of Commerce, National Institute of Standards and Technology.

R. Pass and K. Pietrzak (Eds.): TCC 2020, LNCS 12551, pp. 595–620, 2020.
https://doi.org/10.1007/978-3-030-64378-2_21

obtains the output of the function being computed, then all parties must obtain the output.

It is known how to securely compute every functionality in the above setting, assuming honest majority (i.e. more than half the parties are uncorrupted) [10, 12,17,18,37]. On the other hand, impossibility results, showing that there are n-party functionalities that cannot be computed fairly (even computationally and even with a broadcast channel), are known in the case of no honest majority. Negative results on fairness include the early work of Cleve [13], who showed that fair coin-tossing is impossible when $n/2$ out of n parties are fail-stop (i.e. behave in an honest-but-curious manner with the exception that they may abort early). In the 2-party case, non-trivial functions that can be computed fairly *without* honest majority, were first discovered in the seminal work of Gordon et al. [19]. By now, the 2-party setting is well-understood, with a full characterization of the necessary and sufficient conditions for fair computation of large classes of functionalities [2,3].

In this paper we focus on the n-party case, where $n = \text{poly}(\lambda)$ is any polynomial in λ, the security parameter. We will begin by considering threshold adversaries, these are adversaries who may corrupt up to some threshold th number of parties. In this case, Cleve's result [13] tells us that it is impossible to achieve fairness for all functionalities when $th \geq n/2$.

Threshold Adversaries. The relevant prior works that we are aware of are those of Gordon and Katz [21] and Asharov et al. [3]. Gordon and Katz [21] present fair protocols for 3-party majority and for the OR function (and by symmetry for the AND function) for any polynomial n number of parties and $n-1$ or fewer corruptions. Asharov et al. [3] present n-party protocols with up to $n/2$ corruptions for functions F for which every $n/2$-size partition can be computed fairly in the 2-party setting. We emphasize, however, that the protocol of Asharov et al. [3] only scales to $O(\log \lambda)$ number of parties, regardless of the efficiency of the underlying fair 2-party protocol employed. Moreover, extending their results to more than $n/2$ out of n corruptions was considered an open problem in their work. Thus, prior to our work, AND and OR were the only functionalities for which efficient protocols achieving fairness with no honest majority for any $n = \text{poly}(\lambda)$ parties were known.

We also consider non-threshold adversaries. Specifically, we consider adversarial structures $\mathcal{A}_{\mathsf{adv}}$ for which it is known to be impossible to achieve fairness for all functionalities. We ask whether for such adversarial structures there exist non-trivial functions that can be computed fairly.

Background on MPC with General Adversarial Structures. An adversarial structure $\mathcal{A}_{\mathsf{adv}}$ on a set $[n]$—corresponding to n parties $P_1, \ldots, P_n$—is a monotone collection of non-empty sets S. We say that an MPC protocol is secure with fairness for adversarial structure $\mathcal{A}_{\mathsf{adv}}$ if it is secure with fairness under any set of corruptions $S \in \mathcal{A}_{\mathsf{adv}}$. In the seminal works of Hirt and Maurer [26,27], they defined a set of adversarial structures $Q^{(2)}$, which consists of adversarial

structures $\mathcal{A}_{\mathsf{adv}}$ for which no two sets in $\mathcal{A}_{\mathsf{adv}}$ cover $[n]$. They presented an (inefficient) information-theoretic secure protocol for fail-stop adversaries for adversarial structures $Q^{(2)}$. They also gave a simple argument that it is impossible to achieve (even computational) fairness for adversarial structures not in $Q^{(2)}$ using the classical result of Cleve [13].

Fairness for $(\mathcal{A}_{\mathsf{adv}}, F)$*-pairs.* In this work, we initiate the research direction of achieving MPC protocols with fairness against—possibly non-threshold—adversarial structures $\mathcal{A}_{\mathsf{adv}}$ that are not in $Q^{(2)}$. While for any adversarial structure $\mathcal{A}_{\mathsf{adv}} \notin Q^{(2)}$, it is impossible (even computationally) to achieve MPC with fairness for all functionalities F, there can be some functionalities F for which it is possible to achieve MPC with fairness. We will investigate pairs of functionalities and adversarial structures $(\mathcal{A}_{\mathsf{adv}}, F)$ for which is it possible to achieve fairness in the multiparty setting. To the best of our knowledge, prior work on complete fairness in multiparty computation for adversarial structures outside $Q^{(2)}$ has considered only threshold adversarial structures.

1.1 Our Results

Consider a symmetric Boolean function[1] $F(\boldsymbol{w}) = F(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{w} = \boldsymbol{x}||\boldsymbol{y}$ and $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{n/2}$. We consider n-party MPC protocols for computing the function F. Note that since F is symmetric, there exists a two-input function f such that $F(\boldsymbol{x}, \boldsymbol{y}) = f_{n/2,n/2}(\sum_{i=1}^{n/2} x_i, \sum_{i=1}^{n/2} y_i)$. In our first result, we present a fair MPC protocol for functionalities F that are symmetric and for which the corresponding $f_{n/2,n/2}$ can be computed fairly in the 2-party setting. Importantly, our protocol handles any polynomial $n = \mathrm{poly}(\lambda)$ number of parties (polynomial in security parameter λ) and is secure against $n/2$ or fewer corruptions. Recall that Asharov et al. [3] gave a transformation from fair 2-party protocols to fair n-party protocols, secure against $n/2$ or fewer corruptions. Their transformation, however, requires running the underlying protocol with all possible subsets $S \subseteq [n]$ of size $|S| = n/2$ playing the part of the two parties in the underlying 2-party protocol. This means that their protocol can only handle a number of parties n that is at most logarithmic in the security parameter $n = O(\log(\lambda))$. In this work, we show how to extend their construction to any number of parties n that is polynomial in the security parameter $n = \mathrm{poly}(\lambda)$. However, the extension applies only to symmetric Boolean functions.

Theorem 1 (Informal). *Let $F : \{0,1\}^n \to \{0,1\}$ be a symmetric function, such that there is an efficient protocol for computing $f_{n/2,n/2}$ fairly in the two-party setting. Then for any $n = \mathrm{poly}(\lambda)$, there is an efficient protocol for computing F fairly in the n-party setting with up to $n/2$ corruptions.*

We extend the above result to more general, non-threshold, adversarial structures outside of $Q^{(2)}$, which may include corrupted sets of parties of

[1] In this context, we mean a Boolean function whose output depends only on the number of ones in the input. See [36], Def. 2.8.

size greater than $n/2$. For symmetric F and any $n' \in [n-1]$, we consider $F(\boldsymbol{x}, \boldsymbol{y}) = f_{n',n-n'}(\sum_{i=1}^{n'} x_i, \sum_{i=1}^{n-n'} y_i)$, and require that for all $n' \in [n-1]$ there is an efficient protocol computing $f_{n',n-n'}$ fairly in the two-party setting. For any such F, we define a corresponding set of adversarial structures $Q^{(F)}$. Informally, $Q^{(F)}$ contains adversarial structures $\mathcal{A}_{\mathsf{adv}}$ such that $\mathcal{A}_{\mathsf{adv}}$ can be partitioned into $\mathcal{A}_{\mathsf{adv},1} \in Q^{(2)}$ and $\mathcal{A}_{\mathsf{adv},2} \notin Q^{(2)}$ such that for any pair of distinct sets $(T, T') \in \mathcal{A}_{\mathsf{adv},2}$, $T' \not\subseteq T$. Additionally, we require certain efficient secret sharing schemes corresponding to $\mathcal{A}_{\mathsf{adv},1}$ and $\mathcal{A}_{\mathsf{adv},2}$. See the full version for a formal definition of $Q^{(F)}$.

Theorem 2 (Informal). *Let $F : \{0,1\}^n \to \{0,1\}$ be a symmetric function, such that there is an efficient protocol for computing $f_{n',n-n'}$ fairly in the two-party setting for all $n' \in [n-1]$. Let $Q^{(F)}$ be defined as above. Then for any $n = \mathrm{poly}(\lambda)$, there is an efficient protocol for computing F fairly in the n-party setting under any adversarial structure $\mathcal{A}_{\mathsf{adv}} \in Q^{(F)}$.*

As an additional result of interest, we show that (ignoring efficiency requirements for the underlying secret-sharing schemes) any *projective plane* can be used to construct a non-threshold adversarial structure in $Q^{(F)}$. See the full version for additional details.

In our second main result, we present a fair MPC protocol for the majority function, for any polynomial $n = \mathrm{poly}(\lambda)$ number of parties, and $n/2+1$ or fewer corruptions.

Theorem 3 (Informal). *There is an efficient protocol for computing n-party Majority fairly for any $n = \mathrm{poly}(\lambda)$ (s.t. $n \geq 8$) with $n/2+1$ or fewer corruptions.*

The construction can be straightforwardly extended to work for $n/2 + c$ or fewer corruptions, where c is a constant.

As before, we extend the result to more general, non-threshold, adversarial structures outside of $Q^{(2)}$, by defining a set of adversarial structures $Q^{(\mathsf{Maj})}$. Informally, $Q^{(\mathsf{Maj})}$ contains adversarial structures $\mathcal{A}_{\mathsf{adv}}$ such that $\mathcal{A}_{\mathsf{adv}}$ can be partitioned into $\mathcal{A}_{\mathsf{adv},1} \in Q^{(2)}$ and $\mathcal{A}_{\mathsf{adv},2} \notin Q^{(2)}$ such that for any pair of distinct sets $(T, T') \in \mathcal{A}_{\mathsf{adv},2}$ such that $T' \subseteq T$, it is the case that $|T \backslash T'| \leq c$. Additionally, we require certain efficient secret sharing schemes corresponding to $\mathcal{A}_{\mathsf{adv},1} \in Q^{(2)}$ and $\mathcal{A}_{\mathsf{adv},2} \notin Q^{(2)}$. See the full version for a formal definition of $Q^{(\mathsf{Maj})}$.

Theorem 4 (Informal). *There is an efficient protocol for computing n-party Majority fairly for any $n = \mathrm{poly}(\lambda)$ under every adversarial structure $\mathcal{A}_{\mathsf{adv}} \in Q^{\mathsf{Maj}}$.*

As an additional result of interest, we show that (ignoring efficiency requirements for the underlying secret-sharing schemes) starting from an appropriate type of *combinatorial design* and adding certain sets to it, we obtain a non-threshold adversarial structure in $Q^{(\mathsf{Maj})}$. See the full version for additional details.

1.2 Technical Overview

Half ($n/2$) or Fewer Corruptions. Recall that [3] showed that, for $n = O(\log(\lambda))$ number of parties, a function $F(x_1, \ldots, x_n)$ is computable with fairness under $n/2$ corruptions if and only if for every partition (S_L, S_R) of $[n]$ of size $n/2$, $F([x_i]_{i \in S_L}, [x_i]_{i \in S_R})$ is computable with complete fairness in the two party setting, where one party holds input $[x_i]_{i \in S_L}$ and the other holds input $[x_i]_{i \in S_R}$. We begin by re-casting the protocol of [3] in a player-simulation model (similar to Hirt and Maurer [26,27]). The protocol of [3] considers all possible 2-partition (S_L, S_R) of $[n]$ of size $n/2$ (where S_L always contains 1) and for each partition, parties $P_i, i \in S_L$ simulate virtual party P_L and parties $P_i, i \in S_R$ simulate virtual party P_R in the fair two party protocol Π_F for functionality $F([x_i]_{i \in S_L}, [x_i]_{i \in S_R})$ that exists by assumption. WLOG, we can take the states of P_L and P_R in round r of Π_F to simply consist of "backup values" a^r, b^r, respectively. For simplicity, we first construct an n-party protocol in a "trusted dealer" model (later we will show how to get rid of the assumption). In each round r, the dealer secret shares the backup values of each virtual party P_L (resp. P_R) across the corresponding parties in S_L (resp. S_R). This is referred to as the *inner* secret sharing scheme in [3]. If at any time, all the real parties simulating a certain virtual party (say P_L) abort, the dealer stops handing out shares and the remaining real parties reconstruct virtual P_R's state to obtain the corresponding backup value. The above description relies on the fact that there are at most $n/2$ corruptions, since if exactly $n/2$ parties corresponding to some virtual party P_L abort there is a uniquely identifiable corresponding virtual party P_R, simulated by exactly the remaining set of $n/2$ parties (all of whom are honest). The remaining parties can therefore identify P_R and compute the correct backup value. On the other hand, if the protocol completes, any set of parties of size $n/2$ or more can reconstruct the correct value, since all subsets of size $n/2$ receive the output of the functionality in the final round. To implement the dealer and ensure that the protocol continues if less than $n/2$ parties abort, [3] additionally perform an $(n/2+1)$-out-of-n secret sharing of each real party's state during the preprocessing, called the *outer secret sharing*. In each round, the parties send their share of the outer-secret sharing to each party. In case at most $n/2 - 1$ parties abort, the remaining parties can continue the protocol by simulating the aborting parties using the $(n/2 + 1)$-out-of-n secret sharing. The number of simulated sub-protocols is essentially $\binom{n}{n/2} \approx 2^n/\sqrt{n}$. Thus, they can only handle at most $n = O(\log(\lambda))$ number of parties, where λ is security parameter.

In our first result we show that the above paradigm can be modified to work for symmetric functions $F : \{0,1\}^n \to \{0,1\}$ without requiring the blowup of running the protocol across each possible subset. Since F is symmetric, its value at all inputs is equivalent to the output of some $f_{n/2,n/2} : \{0, \ldots, n/2\} \times \{0, \ldots, n/2\} \to \{0,1\}$. Let us assume that there is a fair protocol $\Pi_{f_{n/2,n/2}}$ for computing $f_{n/2,n/2}$. We describe the constructed fair protocol for n-party functionality F in the "trusted dealer" setting: The dealer receives all the parties' inputs $\boldsymbol{x} = x_1, \ldots, x_n$ and computes $N = \sum_{i=1}^{n} x_i$. For every $z \in \{0, \ldots, n/2\}$,

the dealer runs protocol $\Pi_{f_{n/2,n/2}}(z, N-z)$ and $\Pi_{f_{n/2,n/2}}(N-z, z)$ "in the head" to obtain backup values for each party and each round.[2] Specifically, for virtual party P_L (resp. P_R), its share when running with input z (resp. $N-z$) in the r-th round is denoted $a^{r,z,N-z}$ (resp. $b^{r,z,N-z}$). We now use an appropriate type of secret sharing scheme to share $a^{r,z,N-z}$ (resp. $b^{r,z,N-z}$), which ensures that a set of corrupted parties can open only the backup values corresponding to *one of the virtual parties' views* in a *single execution* of the (at most) $n/2+1$ executions of the underlying 2PC protocol (i.e. corresponding to the view of P_L or P_R in a single $(z, N-z)$ pair). This is done by defining an augmented set $[n] \times \{0,1\}$ and defining access structures over this set. Specifically, a party P_i holding input bit b, will correspond to the element $(i,b) \in [n] \times \{0,1\}$. Thus, parties along with their inputs correspond to subsets S^+ of $[n] \times \{0,1\}$, and a share that a party receives from the dealer depends both on its index i as well as its input b. Let $S^0 := \{(i,0) : i \in [n]\}$ and $S^1 := \{(i,1) : i \in [n]\}$. We will use a secret sharing scheme to share $a^{r,z,N-z}$ (resp. $b^{r,z,N-z}$) so that its value can be reconstructed by any set S^+ that consists of party P_1 holding either input 0 or 1 (resp. does not include party P_1), z (resp. $N-z$) parties holding input 1 (i.e. $|S^+ \cap S^1| \geq z$, resp. $|S^+ \cap S^1| \geq N-z$) and $n/2-z$ (resp. $n/2-(N-z)$) parties holding input 0 (i.e. $|S^+ \cap S^0| \geq n/2-z$, resp. $|S^+ \cap S^0| \geq n/2-(N-z)$). If exactly $n/2$ parties abort, the remaining honest parties output the "backup" value corresponding to *the remaining party* in *the same* underlying protocol execution. E.g., if a set of $n/2$ parties, including P_1, holding z number of 1's, abort, the remaining parties can open $b^{r,z,N-z}$, since if the corrupt parties hold z number of 1's, the honest parties must hold $N-z$ number of 1's and $n/2-(N-z)$ number of 0's. On the other hand, if less than $n/2$ parties abort, the *outer* secret sharing scheme is used to ensure that all the honest parties continue to receive their shares in each round.

Difficulty of a Generic Transformation for more than $n/2$ Corruptions. In the following, we provide some intuition on the difficulty of extending the above protocol to more than $n/2$ corruptions. We do not make any formal claims here. For concreteness, let us assume we want to handle $n/2+1$ corruptions. First, we must ensure that if $n/2+1$ parties abort, the remaining parties can output some backup value from the underlying protocol, as otherwise there is no hope of obtaining a fair protocol. But this means that any set of $n/2-1$ parties must be able to reconstruct a view from the underlying execution, which means that the set of $n/2+1$ corrupted parties will be able to reconstruct *multiple* views (since there are multiple subsets of size $n/2-1$—with distinct values of z—among the set of $n/2+1$ corrupted parties, and each must be able to open an underlying view). When using a generic protocol, it is not clear how to argue that if the underlying protocol is fair when a party sees a single view, it is still fair when a party sees multiple views of the protocol running in parallel with *correlated inputs*. Another difficulty is that if less than $n/2+1$ parties abort—say

[2] If $N-z$ is an invalid input (i.e. $N-z \notin \{0,\ldots,n/2\}$), then the dealer simply uses dummy values.

$n/2-1$ parties abort—then the remaining parties do not necessarily know which backup value to output. As before, there are multiple subsets of size $n/2-1$—with distinct values of z—among the set of $n/2+1$ remaining parties, and each may correspond to a different backup value. Further, note that the outer secret sharing can no longer be used when $n/2-1$ (or more) parties abort, since if the outer secret sharing scheme can be reconstructed by $n/2+1$ or fewer parties, then the set of corrupt parties can recover backup values for round r before round r is executed, thus negating the fairness guarantees of the underlying protocol. Our solution for $n/2+1$ or fewer corruptions will resolve each of these problems, but will use special properties of a specific protocol, and will not work generically for any underlying fair-2-PC protocol.

Direct Construction for Majority with $n/2+1$ or Fewer Corruptions. We next present our protocol for n-party computation of Maj assuming at most $n/2+1$ corruptions. As discussed above, the generic transformation techniques no longer work. Therefore, we extend the two-party protocol of Gordon et al. [19] and the analysis of Asharov et al. [2] to our setting. Specifically, recall that in the 2-party protocol of Gordon et al. [19], the dealer chooses a designated round r^*, drawn from a geometric distribution with parameter α (and with all but negligible probability is assured that $r^* \leq \mathsf{rounds}$, where $\mathsf{rounds} = \omega(\log(\lambda)) \cdot 1/\alpha$ is the number of rounds in the protocol) in which to begin releasing the correct output of the functionality. In the rounds previous to this, each party receives the output of the functionality evaluated with its own input and a randomly chosen input for the other party. Now, in the n party case, we set $R := \{1,2,3\}$. In each round r, the dealer computes backup values $a^{r,R',n',z}$ for each $R' \subseteq R$, $n' \in \{n/2-1, n/2, n/2+1\}$ and each $z \in \{0,\ldots,n'\}$. For $r < r^*$, each value $a^{r,R',n',z}$ is chosen as $f_{n',n-n'}(z,\hat{x})$, where $\hat{x}$ is chosen uniformly from $\{0,\ldots,n-n'\}$, and $f_{n',n-n'}$ outputs 1 when the sum of its inputs is at least $n/2+1$. For $r \geq r^*$, each value $a^{r,R',n',z}$ is set to $f_{n',n-n'}(x,y)$, where x, y are the inputs of the corrupted and uncorrupted parties, respectively. Each $a^{r,R',n',z}$ is shared so that it can be opened by any set S that has a subset W of size n' such that $W \cap R = R'$ and has a subset W' of size n' consisting of z parties holding a 1 input and $n'-z$ parties holding a 0 input. We observe than any set of corrupt parties of size at most $n/2+1$ can open at most a constant number, deg, of backup values. Furthermore, if $n/2-1$ or more parties abort in round r, the remaining set of parties, S', which has size $n' \in \{n/2-1, n/2, n/2+1\}$ and for which $S' \cap R = R'$, run a secure computation protocol (with fairness and guaranteed output delivery, since when $n \geq 8$ we have an honest majority among the remaining parties) to recover $a^{r-1,R',n',z}$ for the appropriate values of R', n', and z. The set R' is needed since in the security proof, we will argue that the backup value opened by the remaining parties cannot be opened by the set of corrupt parties before aborting. If less than $n/2-1$ parties abort, then each remaining party can still recover its share in each round using the outer secret sharing scheme and so the protocol continues. By setting α correctly, the ideal adversary is able to skew the output appropriately (as in [19]), even though the corrupt parties see multiple random values in rounds $r < r^*$. Intuitively, this comes from the fact that the

real adversary will with some $1/\operatorname{poly}(n)$ probability obtain the same view in round r when $r = r^*$ or when $r < r^*$. In the case $r = r^*$, the honest parties output their backup value (which is distributed as described above) in the real world, but always output the correct output value in the ideal world. In the case that $r < r^*$, the honest parties still output their backup value in the real world. However, the simulator in the ideal world can lie about the corrupted parties' inputs and submit values from a carefully constructed distribution to the ideal functionality, since the ideal functionality has not yet been called in the simulation (it is only called in round r^*). Thus, it is possible that for a fixed adversarial view, the distribution of outputs of the honest parties is the same in the real and ideal worlds. To analyze the resulting distributions in the real and ideal world, we follow the techniques of Asharov [2], who explicitly computes the required probabilities as a vector and finds the sufficient conditions so that this vector falls within the convex hull of a set of vectors corresponding to the rows of the truthtable. Unfortunately, the proof of Asharov [2] works only for constant-size domain. Since we want to extend our case to any polynomial number of parties n, we necessarily require a polynomial domain (since the domain will be exactly $\{0, \ldots, n\}$). Specifically, Asharov's technique [2] fixed the domain size to be constant and used existence theorems to prove that α can be set sufficiently small so that the vector is contained in the convex hull. Instead, we consider the spectral norm of the matrix corresponding to the inverse of $\boldsymbol{M}^{+}_{\widetilde{f}}$, where $\boldsymbol{M}_{\widetilde{f}}$ is the truthtable corresponding to a function $\widetilde{f}$ that is closely related to $f_{n',n-n'}$, and $\boldsymbol{M}^{+}_{\widetilde{f}}$ is equal to $\boldsymbol{M}_{\widetilde{f}}$ concatenated with a column of 1's, and show that it is upper bounded by a constant. This allows us to achieve the desired result. We note that the techniques outlined above can be straightforwardly extended to the case of $n/2 + c$ corruptions, where c is a fixed constant.

Extending to more General Adversarial Structures. Secret sharing schemes are used in two ways in the results for threshold adversarial structures described above: (1) The *outer secret sharing scheme*, which ensures that when certain sets of parties abort, the protocol can continue. We require that no set from the adversarial structure is an authorized set for the access structure corresponding to this scheme. (2) The *inner secret sharing scheme*, which ensures that if the surviving parties cannot continue the protocol using the outer secret sharing scheme, they can reconstruct a backup value using this scheme. We can no longer require that no set from the adversarial structure is an authorized set for the access structure corresponding to this scheme. Instead, we merely limit the number of instances of the inner secret sharing scheme that can be opened by the corrupt parties.

To achieve this, we view an arbitrary adversarial structure as the union of a $Q^{(2)}$ adversarial structure, $\mathcal{A}_{\mathsf{adv},1}$ and a non-$Q^{(2)}$ adversarial structure, $\mathcal{A}_{\mathsf{adv},2}$. The outer secret sharing scheme will correspond to access structure, $\mathcal{A}_{\mathsf{hon},1}$, which is equal to the complement of $\mathcal{A}_{\mathsf{adv},1}$. For the inner secret sharing scheme, we consider $\mathcal{A}_{\mathsf{hon},2} = \mathcal{A}_{\mathsf{adv},2}$ and we partition $\mathcal{A}_{\mathsf{hon},2}$ according to the size n' of the authorized sets, yielding sets $\mathcal{A}^{n'}_{\mathsf{hon},2}$. We then obtain monotone access structures

$\mathcal{A}_{\mathsf{hon},2}^{n',+}$ for all $n' \in [n]$, consisting of $\mathcal{A}_{\mathsf{hon},2}^{n'}$ and all supersets of sets in $\mathcal{A}_{\mathsf{hon},2}^{n'}$. We then construct a secret sharing scheme for each n' and each $z \in \{0, \ldots, n'\}$, which allows a set of parties to reconstruct the secret if the set of parties is contained in $\mathcal{A}_{\mathsf{hon},2}^{n',+}$ and the set of parties includes z number of parties holding a 1 and $n' - z$ number of parties holding a 0. For the first result (corresponding to fair computation of symmetric functions) we require that $\mathcal{A}_{\mathsf{adv},2}$ does not contain any two sets T, T' such that $T' \subsetneq T$. For the second result (corresponding to fair computation of Maj) we require that for any two sets $T, T' \in \mathcal{A}_{\mathsf{adv},2}$ if T is a superset of T', it can only contain c additional elements, where c is a constant.

1.3 Related Work

The 2-party Setting. Subsequent to the seminal paper of Gordon et al. [19], a large body of work has been dedicated to understanding which functionalities can be computed fairly in the two-party setting. Various works, culminating in a full characterization for symmetric, constant-size-domain functionalities, include [2,3].

The n-party Setting. Hirt and Maurer [26,27] characterized the set of access structures that are necessary and sufficient for fair n-party computation of *all functionalities*, and dubbed this set $Q^{(2)}$. The question remained of whether there are non-trivial functionalities that can be computed fairly for adversarial structures outside of $Q^{(2)}$. In particular, the works of [21] and [3], which have already been discussed above, considered threshold access structures outside of $Q^{(2)}$.

Partial Fairness and Other Notions. Another line of works has considered achieving partial fairness (also called $1/p$-fairness) guarantees for large classes of functionalities, even when there is no honest majority. Specifically, the goal is to obtain protocols for which the real and ideal world are distinguishable by at most $1/p$, for some polynomial $p = p(\lambda)$. Partial fairness has been studied in both the 2-party and multiparty setting [7,9,22]. Note that our focus in the current work is to achieve "complete" fairness, where the real and ideal world are computationally indistinguishable. "Best of both worlds" security has also been studied–where protocols are required to achieve fairness in the case of honest majority and security-with-abort in the case of honest minority [7,28,29]. We also mention other desirable security properties related to fairness that have been considered in the literature such as *guaranteed output delivery* [14,23] and *security with identifiable abort* [30,31].

Partial Fairness for Coin-Tossing. For the special case of coin-tossing, it is known by the classical result of Cleve [13] that complete fairness is impossible. However, there are several results in the two-party and multi-party settings that deal with achieving partial fairness—i.e. bias of $1/p$—for the best possible p [1,6,8,11,34,35].

Lower Bounds. Lower bounds on number of rounds or computational assumptions necessary to achieve (partially) fair protocols have also been studied [15,16,24,25]. Complete primitives for fairness and primitives that imply

secure coin-tossing were studied in [4,20]. Further works have elucidated properties of protocols necessary to achieve fairness [33].

2 Notation, Definitions and Preliminaries

Definitions of MPC with full security (i.e. fairness) and security-with-abort are deferred to the full version. We follow [5] for the definitions of access structures and secret sharing schemes. Given a set $S \subseteq [n]$, denote by $\overline{S} := [n] \setminus S$ and by $\mathcal{P}(S)$ the power set of S.

Useful Access Structures. We consider access structures over the set $[n]$, as well as the set $[n]\times\{0,1\}$. Let $S^0 := [n]\times\{0\}$ and $\mathcal{S}^1 := [n]\times\{1\}$. Access structure $\mathcal{A}_{a,z,n'-z,2n}$, for $n' \in [n]$ and $z \in \{0,\ldots,n'\}$, consists of sets $S^+ \subseteq [n] \times \{0,1\}$ with corresponding $S := \{i : (i,0) \text{ or } (i,1) \in S^+\}$ that satisfy all of the following: (1) $1 \in S$; (2) $|S^+ \cap S^1| \geq z$; (3) $|S^+ \cap S^0| \geq n' - z$.

Access structure $\mathcal{A}_{b,z,n'-z,2n}$, for $n' \in [n]$ and $z \in \{0,\ldots,n'\}$, consists of sets $S^+ \subseteq [n] \times \{0,1\}$ with corresponding $S := \{i : (i,0) \text{ or } (i,1) \in S^+\}$ that satisfy all of the following: (1) $1 \notin S$; (2) $|S^+ \cap S^1| \geq z$; (3) $|S^+ \cap S^0| \geq n' - z$.

More generally, let R be a set of distinguished elements of $[n]$. Let $R' \subseteq R$ and let $R'' = R \setminus R'$. Access structure $\mathcal{A}_{R,R',z,n',2n}$, for $R' \subseteq R$, $n' \in [n]$ and $z \in \{0,\ldots,n'\}$, consists of sets $S^+ \subseteq [n] \times \{0,1\}$ with corresponding $S := \{i : (i,0) \text{ or } (i,1) \in S^+\}$ that satisfy all of the following: (1) $R' \subseteq S$; (2) $R'' \cap S = \emptyset$; (3) $|S^+ \cap S^1| \geq z$; (4) $|S^+ \cap S^0| \geq n' - z$.

See full version for constructions.

3 Symmetric Functions and $n/2$ Corruptions

Let $F : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ be a symmetric Boolean function. Then we have that for all $\boldsymbol{x} \in \{0,1\}^{n/2}$ and $\boldsymbol{y} \in \{0,1\}^{n/2}$, $F(\boldsymbol{x},\boldsymbol{y}) = f_{n/2,n/2}(\sum_{i=1}^{n/2} x_i, \sum_{i=1}^{n/2} y_i)$, for some $f_{n/2,n/2}$. Assume that $f_{n/2,n/2} : \{0,\ldots,n/2\} \times \{0,\ldots,n/2\}$ can be fairly computed in the two-party setting and let $\Pi_{f_{n/2,n/2}}$ denote the two-party protocol (with parties P_L, P_R) that fairly computes $f_{n/2,n/2}(x,y)$. For $x,y \in \{0,\ldots,n/2\}$, let $\Pi_{f_{n/2,n/2}}(x,y)$ denote an execution of $\Pi_{f_{n/2,n/2}}$, where P_L has input x and P_R has input y. Let $a^{x,y,r}_{f_{n/2,n/2}}$ denote the backup value of P_L in the r-th round of an execution of $\Pi_{f_{n/2,n/2}}(x,y)$ and let $b^{x,y,r}_{f_{n/2,n/2}}$ denote the backup value of P_R in the r-th round of the same execution of $\Pi_{f_{n/2,n/2}}(x,y)$. In the following, p is set to $p = 2 \cdot (n/2+1)$.

Theorem 5. *Let F, $f_{n/2,n/2}$ be as above. Assume there is an efficient protocol for computing $f_{n/2,n/2}$ fairly in the two-party setting. Then for any $n = \text{poly}(\lambda)$, the protocol presented in Fig. 1 (and Fig. 2) is an efficient protocol for computing F fairly in the n-party setting with $n/2$ or fewer corruptions.*

The protocol in Fig. 1 uses a secret sharing scheme for access structure $\mathcal{A}_{a,z,n/2-z,2n}$ and $\mathcal{A}_{b,z,n/2-z,2n}$, defined in Sect. 2.

1. The parties $P_1, \ldots, P_n$ hand their inputs, denoted $\boldsymbol{x} =_1, \ldots, _n \in \{0,1\}^n$, respectively, to the dealer. If a party P_j does not send an input, then the dealer selects $_j \in \{0,1\}$ uniformly at random. If half of the parties do not send an input, then the dealer sends $f_{n/2,n/2}(\sum_{i=1}^{n/2} {}_i, \sum_{i=n/2+1}^{n} {}_i)$ to the honest parties and halts. Let $N := \sum_{i=1}^{n} {}_i$.
2. The dealer computes for $z \in \{\max\{0, N - n/2\}, \ldots, \min\{N, n/2\}\}$, $r \in [\mathsf{rounds}]$, the backup outputs $a^{z,r} := a^{z,N-z,r}_{f_{n/2,n/2}} || 0^\lambda$, $b^{N-z,r} := b^{z,N-z,r}_{f_{n/2,n/2}} || 0^\lambda$ for an execution of $\Pi_{f_{n/2,n/2}}(z, N-z)$. The dealer also sends back to each P_i an authentication of its input $_i$.
3. If $N > n/2$, then for $z \in \{0, \ldots, N - n/2 - 1\}$, $r \in [\mathsf{rounds}]$ set $a^{z,r} := \mathbf{0}$, $b^{z,r} := \mathbf{0}$.
4. If $N < n/2$, then for $z \in \{N+1, \ldots, n/2\}$, $r \in [\mathsf{rounds}]$, set $a^{z,r} := \mathbf{0}$, $b^{r,z} := \mathbf{0}$.
5. For $r \in [\mathsf{rounds}]$,
   (a) For $z \in \{0, \ldots, n/2\}$, the dealer secret shares $a^{z,r}$ using access structure $\mathcal{A}_{a,z,n/2-z,2n}$, producing (authenticated) shares $[\tilde{\boldsymbol{s}}_i^{b,z,r}]_{b\in\{0,1\},i\in[n]}$. Each party P_i holding input b receives shares $[\tilde{\boldsymbol{s}}_i^{b,z,r}]_{z\in\{0,\ldots,n/2\}}$. If $n/2$ parties abort, including P_1, then the remaining parties (corresponding to set S) submit their (authenticated) inputs and shares from round $r-1$ to $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$, output whatever it outputs and halt. Note that all parties in S are honest.
   (b) For $z \in \{0, \ldots, n/2\}$ the dealer secret shares $b^{r,z}$ using access structure $\mathcal{A}_{b,z,n/2-z}$, producing (authenticated) shares $[\overline{\boldsymbol{s}}_i^{b,z,r}]_{b\in\{0,1\},i\in[n]}$. Each party P_i holding input b receives shares $[\overline{\boldsymbol{s}}_i^{b,z,r}]_{z\in\{0,\ldots,n/2\}}$. If $n/2$ parties abort, not including P_1, then the remaining parties (corresponding to set S) submit their (authenticated) inputs and shares from round r to ideal functionality $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$, output whatever it outputs and halt. Note that all parties in S are honest.
6. Otherwise, the remaining parties (set S) submit their (authenticated) inputs and shares to ideal functionality $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$, output whatever it outputs and halt. If some set $\widetilde{S}$ of parties abort, preventing the remaining parties from receiving output, the remaining parties: $S := S \setminus \widetilde{S}$ go back to the beginning of Step 6 and resubmit their shares to $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$. This continues until the honest parties receive the output from the ideal functionality.

Fig. 1. Fair, efficient, multiparty computation of F with n parties and $n/2$ or fewer corruptions.

Functionality $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$

- **Input:** Set $S \subseteq [n]$ of size n'. For $i \in S$, the i-th party's input is (authenticated) bit $_i$ and p shares $[\tilde{\boldsymbol{s}}^k_i]_{k\in[p]}$. Let S' be the set of parties who submit input shares that are properly authenticated. We have that $|S'| \geq n/2$. We also assume WLOG that $|S'| = n/2$. If it is greater, then we just compute with some subset of the input shares. Let $z := \sum_{i\in S'} {}_i$.
- **Function Computation:** If $1 \in S'$ (resp. $1 \notin S'$), then for $k \in [p]$, run reconstruction algorithm Recon for secret-sharing scheme $\mathcal{A}_{a,z,n'-z,2n}$ (resp. $\mathcal{A}_{b,z,n'-z,2n}$) with input shares $[\tilde{\boldsymbol{s}}^k_i]_{i\in S'}$. to obtain candidates $[\mathsf{secret}^k = \mathsf{secret}^k_1||\mathsf{secret}^k_2]_{k\in[p]}$.
- **Output:** secret^k_1 corresponding to the first $k \in [p]$ such that $\mathsf{secret}^k_2 = 0^\lambda$.

Fig. 2. Reconstruction functionality with respect to p sets of secret shares.

Proof. Let $T \subseteq [n], |T| = n/2$ denote the set of corrupt parties. Assume WLOG that $1 \in T$. Sim applies the simulator $\mathsf{Sim}_{f_{n/2,n/2}}$ of the two-party protocol $\Pi_{f_{n/2,n/2}}$.

- Sim constructs the following adversary $A_{f_{n/2,n/2}}$ for $\Pi_{f_{n/2,n/2}}$, playing the same role as A.
  - $A_{f_{n/2,n/2}}$ invokes A expecting its inputs $\boldsymbol{x}$.
  - $A_{f_{n/2,n/2}}$ sends inputs $x = \sum_{i\in T} x_i$ to the dealer of Π_f.
  - For $r = 1, \ldots, \mathsf{rounds}$, upon receiving backup value a^r, set $a^{x,r} = a^r||0^\lambda$ and $a^{z,r} = \mathbf{0}$ for $z \in \{0, \ldots, n/2\} \setminus \{x\}$. For $z \in \{0, \ldots, n/2\}$, secret share $a^{z,r}$ using access structure $\mathcal{A}_{a,z,n/2-z,2n}$, producing shares $[\tilde{\boldsymbol{s}}^{b,z,r}_i]_{b\in\{0,1\},i\in[n]}$. Each party P_i holding input b receives shares $[\tilde{\boldsymbol{s}}^{b,z,r}_i]_{z\in\{0,\ldots,n/2\}}$.
  - If all the parties in T abort, then $A_{f_{n/2,n/2}}$ aborts, otherwise it continues.
  - If the final round rounds completes, $A_{f_{n/2,n/2}}$ submits shares for all remaining parties in T to the ideal functionality and simulates an output of out in return.
- Let $\mathsf{Sim}_{f_{n/2,n/2}}$ be the simulator for $A_{f_{n/2,n/2}}$ in the hybrid model.
- The simulator Sim interacts with the two-party protocol simulator $\mathsf{Sim}_{f_{n/2,n/2}}$ by invoking it on adversary A_f with input x. It then receives a simulated view for $A_{f_{n/2,n/2}}$, containing its random coins and backup outputs. Having received this view of $A_{f_{n/2,n/2}}$, the simulator S_f can extract from it the view of A in this execution, as it is implied by the view of $A_{f_{n/2,n/2}}$. Specifically, the randomness $A_{f_{n/2,n/2}}$ uses to share different secrets determines the shares that the corrupted parties see. If $A_{f_{n/2,n/2}}$ does not abort before the final reconstruction, $\mathsf{Sim}_{f_{n/2,n/2}}$ obtains from $A_{f_{n/2,n/2}}$'s view any inputs to the functionality $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$. It uses the output out contained in the view (since the last round was reached) to simulate the output of the ideal functionality $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$. If some parties abort and the remaining parties re-submit their

inputs to the ideal functionality, $\mathsf{Sim}_{f_{n/2,n/2}}$ can still use out to simulate the output each time.

3.1 Implementing the Dealer and $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$

This is done similarly to Asharov et al. [3] and our exposition follows theirs. Following [3,7,8], we eliminate the trusted on-line dealer of our multiparty protocols in a few steps using a few layers of secret-sharing schemes. In the first step, we convert the on-line dealer to an off-line dealer. That is, we construct a protocol in which the dealer sends only one message to each party in an initialization stage; the parties then interact in rounds using a broadcast channel (without the dealer) and in each sub-round of round i each party learns its shares of the r-th round. Specifically, in round r, party P_j learns a share in a secret sharing scheme for access structure $\mathcal{A}_{a,z,n/2-z,2n}$, $\mathcal{A}_{b,z,n/2-z,2n}$, for every $z \in \{0,\ldots,n/2\}$ (we call these shares P_j's shares of the inner secret-sharing scheme).

For this purpose, the dealer computes, in a preprocessing phase, the appropriate shares for the inner secret-sharing scheme. For each round, the shares of each party P_j are shared in a *special* 2-out-of-2 secret-sharing scheme, where P_j gets one of the two shares (called the mask). In addition, all parties (including P_j) receive shares in a $n/2+1$-out-of-n secret-sharing scheme of the other share of the 2-out-of-2 secret sharing. We call the resulting secret-sharing scheme the *outer* $(n/2+1)$-out-of-n scheme ($n/2$ parties and the holder of the mask are needed to reconstruct the secret).

The use of the outer secret-sharing scheme with threshold $n/2+1$ plays a crucial role in eliminating the on-line dealer. On one hand, it guarantees that an adversary, corrupting at most $n/2$ parties cannot reconstruct the shares of round r before round r. On the other hand, at least $n/2$ parties must abort to prevent the reconstruction of the outer secret-sharing scheme. Note that $n/2$ aborting parties can prevent the remaining parties from receiving their shares and, indeed, in the description of the protocol, if $n/2$ parties abort, the remaining parties no longer receive shares from the dealer. Finally, we replace the off-line dealer by using a secure-with-abort and cheat-detection protocol computing the functionality computed by the dealer.

To prevent corrupted parties from cheating, by e.g., sending false shares and causing reconstruction of wrong secrets, every message that a party should send during (any possible flow of) the execution of the protocol is signed in the preprocessing phase (together with the appropriate round number and the party's index). In addition, the dealer sends a verification key to each of the parties. To conclude, the off-line dealer gives each party the signed shares for the outer secret-sharing scheme together with the verification key.

Whenever $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$ is run in Steps 5a and 5b, all parties are honest, so it can be trivially implemented. When $F^{\mathsf{th},n/2}_{\mathsf{Recon},S,p}$ is run in Step 6, there may *not* be an honest majority. In this case, however, it is the final round so the reconstruction protocol will output the same value, regardless of which subset of parties participate (as long as the subset includes all the $n/2$ honest parties). Thus,

the adversary may get its output early and abort to prevent the honest parties to obtain output. The view of the adversary can be simulated since the ideal functionality has already been called at this time. Moreover, the protocol simply gets restarted until either no party aborts during the protocol (which happens in the worst case when only honest parties are remaining).[3] Therefore, the honest parties are guaranteed to obtain their output. We emphasize that the ideal functionality checks that the shares inputted by the parties are correctly authenticated (and are those same shares that were distributed by the "dealer"). Note also that corrupt parties may input an incorrect verification key for verifying the authenticated inputs and shares. In this case, the MPC functionality will partition the inputs according to the submitted verification key. Each party will receive as output the evaluation of the functionality with respect to the inputs of the set of parties who inputted the same verification key as it did.

4 Majority and $n/2+1$ Corruptions

We begin by presenting the protocol for computing n-party majority (Maj) in Figs. 3 and 4. The protocol in Fig. 3 uses a secret sharing scheme for access structure $\mathcal{A}_{R,R',z,n'-z,2n}$, defined in Sect. 2. In the following, p is set to $p = 8 \cdot (3n/2+3)$.

Notation. Let T be the set of corrupted parties with corresponding input $\boldsymbol{x}$, where $\boldsymbol{x}$ is indexed by the elements of T. Let $\overline{T} = [n] \backslash T$ be the set of uncorrupted parties with corresponding input $\boldsymbol{y}$, where $\boldsymbol{y}$ is indexed by the elements of $\overline{T}$. Let $x := \sum_{i \in T} x_i$ and $y := \sum_{i \in \overline{T}} y_i$. Let $T' \subseteq T$, $|T'| \geq n/2 - 1$ be the subset of parties who do not submit valid inputs in Step 4. Let $x^+ = \sum_{i \in T'} x_i$, $x^- = \sum_{i \in T \backslash T'} x_i$.

Define $f^{\mathsf{val}}_{n_1,n_2}(x,y)$ where $n_1 + n_2 = n$, $x \in \{0,\ldots,n_1\}, y \in \{0,\ldots,n_2\}$ and $\mathsf{val} \in \{0,\ldots,2\}$ to be the function that outputs 1 if $x + y + \mathsf{val} \geq n/2 + 1$ and outputs 0 otherwise. If $\mathsf{val} = 0$, we sometimes abbreviate by $f_{n_1,n_2}(x,y) = f^{\mathsf{val}}_{n_1,n_2}(x,y)$. Let $\boldsymbol{M}_{f^{\mathsf{val}}_{n_1,n_2}}$ be the truth table corresponding to $f^{\mathsf{val}}_{n_1,n_2}$. Define the distribution $\boldsymbol{X}_{Real,m}$ to be the uniform distribution over $\{0,\ldots,m\}$.

Let $\boldsymbol{a}$ be a vector of length $4n+12$, indexed by tuples (R',n',z), where $R' \subseteq R = \{1,2,3\}$, $n' \in \{n/2-1, n/2, n/2+1\}, z \in \{0,\ldots,n'\}$. On input $\boldsymbol{x}$, We define a function $\phi(\boldsymbol{x})$ that outputs a set of triples (R',n',z), such that $(R',n',z) \in \phi(\boldsymbol{x})$ if there exists a subset $W \subseteq T$ of size $|W| = n'$ such that $W \cap R = R'$ and a subset $W' \subseteq T$ of size $|W'| = n'$ such that $z = \sum_{i \in W'} x_i$. For any set T of size $|T| \leq n/2+1$ and input $\boldsymbol{x} \in \{0,1\}^{|T|}$, $|\phi(\boldsymbol{x})|$ is at most

[3] We require an *identifiable abort* property to allow elimination of aborting/misbehaving parties and restarting of the protocol. Similar properties were needed in the work of [21]. They required secure computation with designated abort: If the output of the protocol is $\perp$, the parties restart without the lowest indexed party. Also, if the protocol outputs a set $\mathcal{S}$ (indicating those parties whose inputs were inconsistent), the set $\mathcal{S}$ is eliminated.

1. The parties $P_1, \ldots, P_n$ hand their inputs, denoted $\boldsymbol{x} = x_1, \ldots, x_n \in \{0,1\}^n$, respectively, to the dealer. If a party P_j does not send an input, then the dealer selects $x_j \in \{0,1\}$ uniformly at random. If $n/2+1$ of the parties do not send an input, then the dealer sends $f_{n/2-1,n/2+1}(\sum_{i=1}^{n/2-1} x_i, \sum_{i=n/2}^{n} x_i)$ to the honest parties and halts.
2. The dealer chooses a r^* from a geometric distribution with parameter α. For $r \in [\mathsf{rounds}]$, $r < r^*$, $n' \in \{n/2-1, n/2, n/2+1\}$, $z \in \{0, \ldots, n'\}$ and for each subset $R' \subseteq R = \{1,2,3\}$, sample $\hat{x} \sim \boldsymbol{X}_{real,n-n'}$ and set $a^{r,R',n',z} := f_{n',n-n'}(z, \hat{x})$. For $r \in [\mathsf{rounds}]$, $r \geq r^*$, $n' \in \{n/2-1, n/2, n/2+1\}$, $z \in \{0, \ldots, n'\}$, set $a^{r,R',n',z} := \mathsf{out}$, where out denotes the output of the Majority function.
3. For $r \in [\mathsf{rounds}]$, $r < r^*$, for $n' \in \{n/2-1, n/2, n/2+1\}$, $z \in \{0, \ldots, n'\}$, and for each subset $R' \subseteq R = \{1,2,3\}$, the dealer secret shares $a^{r,R',n',z}||0^n$ using access structure $\mathcal{A}_{R,R',z,n'-z,2n}$, producing shares $[\tilde{\boldsymbol{s}}_i^{b,R,R',n',z}]_{b\in\{0,1\},i\in[n]}$. Each party P_i holding input b receives shares $[\tilde{\boldsymbol{s}}_i^{b,R,R',n',z}]$.
4. If $n/2-1$ or more parties abort, then the remaining parties (corresponding to set S) submit their shares from round $r-1$ to ideal functionality $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$, output whatever it outputs and halt. Let S' denote the set of parties who submit properly authenticated shares to $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$. Note that S and S' contain an honest majority.
5. Otherwise, the remaining parties (set S) submit their final shares to ideal functionality $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$, output whatever it outputs and halt. If some set $\widetilde{T}$ of parties abort, preventing the remaining parties from receiving output, the remaining parties: $S := S \setminus \widetilde{T}$ go back to the beginning of Step 5 and resubmit their final shares to $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$. This continues until the honest parties receive the output.

Fig. 3. Fair, efficient, multiparty computation of Maj with n parties and $n/2+1$ or fewer corruptions.

Functionality $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$

- **Input Stage:** Set $S \subseteq [n]$ of size n'. For $i \in S$, the i-th party's input is (authenticated) bit x_i and p shares $[\tilde{\boldsymbol{s}}_i^k]_{k\in[p]}$. Let S' be the set of parties who submit input shares that are properly authenticated. Let $R' = S' \cap R$. We have that $|S'| \geq n/2-1$. We also assume WLOG that $|S'| \leq n/2+1$. If it is greater, then compute with some subset of the input shares. Let $z := \sum_{i\in S'} x_i$.
- **Function Computation:** For $k \in [p]$, run reconstruction algorithm Recon for secret-sharing scheme $\mathcal{A}_{R,R',z,n'-z,2n}$ with input shares $[\tilde{\boldsymbol{s}}_i^k]_{i\in S'}$. to obtain candidates $[\mathsf{secret}^k = \mathsf{secret}_1^k || \mathsf{secret}_2^k]_{k\in[p]}$.
- **Output:** secret_1^k corresponding to the first $k \in [p]$ such that $\mathsf{secret}_2^k = 0^\lambda$.

Fig. 4. Reconstruction functionality with respect to p sets of secret shares.

a constant, deg, where $\mathsf{deg} \leq 3 \cdot 8 \cdot 3 = 48$. Define $\boldsymbol{a}_0$ (resp. $\boldsymbol{a}_1$) such that all indeces in $\phi(\boldsymbol{x})$ are set to 0 (resp. 1) and all other indeces are set to $\perp$.

For $(R', n', z) \in \phi(\boldsymbol{x})$, define $p^{R',n',z}(x) := \Pr_{\hat{y} \sim \{0,\ldots,n-n'\}}[f_{n',n-n'}(z, \hat{y}) = 1]$ and $\overline{p}^{R',n',z}(x) := 1 - p^{R',n',z}(x)$. $p^{R',n',z}(x)$ denotes the probability that the corrupt parties, using sets $W, W' \subseteq T$, where $W \cap R = R'$, $|W'| = n'$, and $z = \sum_{i \in W'} x_i$, reconstruct a 1. For $(R', n', z) \notin \phi(\boldsymbol{x})$, define $p^{R',n',z}(x) := 1$ and $\overline{p}^{R',n',z}(x) := 1$.

Definition 1. *We say that a setting of parameters* $(T, t, \boldsymbol{x}, T', t', x^+, x^-, \boldsymbol{a})$ *is* valid *if:*

1. $T \subseteq [n]$, $n/2 - 1 \leq |T| = t \leq n/2 + 1$.
2. $T' \subseteq T$, $|T'| = t' \geq n/2 - 1$.
3. $\boldsymbol{x} \in \{0,1\}^{|T|}$
4. $x^+ = \sum_{i \in T'} x_i$. $x^- = \sum_{i \in T \setminus T'} x_i$,
5. *Indeces of* $\boldsymbol{a}$ *in* $\phi(\boldsymbol{x})$ *are set to* $0 \backslash 1$ *and all other indeces are set to* $\perp$.

We say that a setting of parameters $(T, t, \boldsymbol{x}, T', t', x^+, x^-)$ *is* valid *if all the above except (5) hold.*

For every valid $(T, t, \boldsymbol{x}, T', t', x^+, x^-)$, for $k \in \{0, \ldots, n-t\}$, define the probabilities $p_{y=k}^{x^-,t,t'} := \Pr_{\hat{x} \sim \boldsymbol{X}_{Real,t'}}[f_{t',n-t}^{x^-}(\hat{x}, y = k) = 1]$. $p_{y=k}^{x^-,t,t'}$ corresponds to the probability the honest parties output a 1 in the Real execution in rounds prior to the designated round r^*, when the combined input of the honest parties is $y = k$, the input of the t' aborting parties is chosen from $\boldsymbol{X}_{Real,t'}$, and the input of the $(t - t')$ corrupt but non-aborting parties is x^-.

For every valid $(T, t, \boldsymbol{x}, T', t', x^+, x^-)$, define the row vectors $\boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}_0} = (q_{y=n-t}^{x^+,x^-,\boldsymbol{a}_0}, \ldots, q_{y=0}^{x^+,x^-,\boldsymbol{a}_0})$ and $\boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}_1} = (q_{y=n-t}^{x^+,x^-,\boldsymbol{a}_1}, \ldots, q_{y=0}^{x^+,x^-,\boldsymbol{a}_1})$ indexed by $k \in \{0, \ldots, n-t\}$ as follows:

$$q_{y=k}^{x^+,x^-,\boldsymbol{a}_0} = \begin{cases} p_{y=k}^{x^-,t,t'} \text{ if } f_{t',n-t}^{x^-}(x^+, y = k) = 1 \\ p_{y=k}^{x^-,t,t'} + \frac{\alpha \cdot p_{y=k}^{x^-,t,t'}}{(1-\alpha) \cdot \prod_{(R',n',z)} (\overline{p}^{R',n',z}(x))}) \text{ if } f_{t',n-t}^{x^-}(x^+, y = k) = 0 \end{cases}$$

$$q_{y=k}^{x^+,x^-,\boldsymbol{a}_1} = \begin{cases} p_{y=k}^{x^-,t,t'} \text{ if } f_{t',n-t}^{x^-}(x^+, y = k) = 0 \\ p_{y=k}^{x^-,t,t'} + \frac{\alpha \cdot (p_{y=k}^{x^-,t,t'} - 1)}{(1-\alpha) \cdot \prod_{(R',n',z)} p^{R',n',z}(x)}) \text{ if } f_{t',n-t}^{x^-}(x^+, y = k) = 1 \end{cases}$$

For every valid $(T, t, \boldsymbol{x}, T', t', \boldsymbol{a}, x^+, x^-)$, such that $\boldsymbol{a} \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$, define the row vectors $\boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}} = (q_{y=n-t}^{x^+,x^-,\boldsymbol{a}}, \ldots, q_{y=0}^{x^+,x^-,\boldsymbol{a}})$, indexed by $k \in \{0, \ldots, n-t\}$ as follows: $\boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}} = (p_{y=n-t}^{x^-,t,t'}, \ldots, p_{y=0}^{x^-,t,t'})$.

Intuition. $q_{y=k}^{x^+,x^-,\boldsymbol{a}}$ corresponds to the probability that the Ideal honest parties receive an output of 1, when the simulator chooses its input to the Ideal functionality from distribution $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}}$,in the case that the adversary aborts

in a round prior to the designated round r^*, the honest parties collectively hold input $y = k$, the aborting parties hold input x^+, the corrupted but non-aborting parties hold input x^-, and the view of the adversary consists of $\boldsymbol{a}$. Our goal is to set the values of $q_{y=k}^{x^+,x^-,\boldsymbol{a}_0}$ so that the distributions in the Ideal and Real world are identical. Note, however, that the simulator does not know the value of y. Therefore, the simulator can only sample from a single probability distribution for all possible values of y, denoted $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}}$, and we must ensure that the resulting distribution over outputs, corresponding to $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} \cdot \boldsymbol{M}_{f_{t',n-t}^{x^-}}$, produces the desired values of $\boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}} = (q_{y=n-t}^{x^+,x^-,\boldsymbol{a}}, \ldots, q_{y=0}^{x^+,x^-,\boldsymbol{a}})$.

In the upcoming theorem, we show that setting $\boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}} = (q_{y=n-t}^{x^+,x^-,\boldsymbol{a}}, \ldots, q_{y=0}^{x^+,x^-,\boldsymbol{a}})$ as described above, yields identical distributions in the Ideal/Real worlds. Then, we must show that there exists a probability vector $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}}$ such that $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} \cdot \boldsymbol{M}_{f_{t',n-t}^{x^-}} = \boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}}$.

We observe that in some cases finding $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}}$ as above is easy. Specifically, for every valid $(T, t, \boldsymbol{x}, T', t', \boldsymbol{a}, x^+, x^-)$, and for $\boldsymbol{a} \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$, $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} = \boldsymbol{X}_{real,t'}$ satisfies $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} \cdot \boldsymbol{M}_{f_{t',n-t}^{x^-}} = \boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}}$.

Theorem 6. *Assume that for every valid setting of parameters* $(T, t, \boldsymbol{x}, T', t', x^+, x^-, \boldsymbol{a})$*, there exists a probability vector* $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}}$ *such that*

$$\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} \cdot \boldsymbol{M}_{f_{t',n-t}^{x^-}} = \boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}}.$$

Then the protocol in Fig. 3 securely computes Maj *for any* $n = \mathrm{poly}(\lambda)$ *(s.t.* $n \geq 8$*) and* $|T| \leq n/2 + 1$ *corruptions.*

Proof. We begin with a description of the simulator Sim:

- Sim invokes A expecting its inputs $\boldsymbol{x}$, as sent to the dealer.
- Sim samples r^* from a geometric distribution with parameter α.
- For every $r = 1$ to $r^* - 1$
  - For $n' \in \{n/2 - 1, n/2, n/2 + 1\}$, $z \in \{0, \ldots, n'\}$, and $R' \subseteq R$, sample $\hat{x} \sim \boldsymbol{X}_{real,n-n'}$ and set $a^{r,R',n',z} := f_{n',n-n'}(z, \hat{x})$. Secret share each $a^{r,R',n',z}||0^n$ using access structure $\mathcal{A}_{R,R',z,n'-z,2n}$, producing shares $[\tilde{\boldsymbol{s}}_i^{b,R',n',z,2n}]_{b\in\{0,1\},i\in[n]}$. Each party $P_i \in T$ holding input b receives shares $[\tilde{\boldsymbol{s}}_i^{b,R',n',z,2n}]$.
  - Fix the resulting view $\boldsymbol{a}$, consisting of the $a^{r,R',n',z}$ values that can be reconstructed by the adversary holding input $\boldsymbol{x}$.
  - If $n/2 - 1$ parties abort, Sim simulates the ideal functionality $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$. Recall that $S' \subseteq S$ submit valid inputs for $F_{\mathsf{Recon},S,p}^{\mathsf{th},n/2+1}$ to Sim. Let $T' = [n] \setminus S'$, where $|T'| = t'$. Let $x^+ = \sum_{i\in T'} \boldsymbol{x}_i$ and $x^- = \sum_{i\in T\setminus T'} \boldsymbol{x}_i$. Sim chooses $\hat{x} \sim \boldsymbol{X}_{Ideal,t'}^{x^+,x^-,\boldsymbol{a}}$ and submits $\hat{x} + x^-$ to the ideal functionality,

receiving out in return. Note that the set S enjoys an honest majority, and so we can compute $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$ with fairness and guaranteed output delivery. Sim returns out as the output of $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$.

- For $r = r^*$
  - Sim sends input x to the ideal functionality computing $f_{t,n-t}$ and receives $\mathsf{out} = f_{t,n-t}(x,y)$. For $n' \in \{n/2-1, n/2, n/2+1\}$, $z \in \{0,\ldots,n'\}$, and $R' \subseteq R$, set $a^{r,R',n',z} := \mathsf{out}$. Secret share each $a^{r,R',n',z}||0^n$ using access structure $\mathcal{A}_{R,R',z,n'-z,2n}$, producing shares $[\tilde{\boldsymbol{s}}_i^{b,R',n',z,2n}]_{b\in\{0,1\},i\in[n]}$. Each corrupt party $P_i \in T$ holding input b receives shares $[\tilde{\boldsymbol{s}}_i^{b,R',n',z,2n}]$.
- For $r > r^*$
  - For $n' \in \{n/2-1, n/2, n/2+1\}$, $z \in \{0,\ldots,n'\}$ and $R' \subseteq R$, set $a^{r,R',n',z} := \mathsf{out}$. Secret share each $a^{r,R',n',z}||0^n$ using access structure $\mathcal{A}_{R,R',z,n'-z,2n}$, producing shares $[\tilde{\boldsymbol{s}}_i^{b,R',n',z,2n}]_{b\in\{0,1\},v\in[n]}$. Each party P_i holding input b receives shares $[\tilde{\boldsymbol{s}}_i^{b,R',n',z,2n}]$.
- Final share reconstruction. At this point, Sim holds the output out from the ideal functionality. Furthermore, the same out will be reconstructed by any set of parties of size $n/2-1$ or more that remain. Sim also obtains from A any inputs to the functionality $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$ in the last stage. It uses out to simulate the output of the ideal functionality $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$. If some parties abort and the remaining parties re-submit their inputs to the ideal functionality, Sim_f can still use out to simulate the output each time.

In case the adversary aborts exactly at r^*, the simulator Sim sends the input x to the trusted party, and so both parties receive $f_{t,n-t}(x,y)$, unlike the real execution. Moreover, in case the adversary has aborted at round $r < r^*$, upon viewing $\boldsymbol{a}$ at round i, the simulator Sim chooses input $\hat{x}$ according to distribution $\boldsymbol{X}^{x^+,x^-,\boldsymbol{a}}_{ideal,t'}$ and submits $\hat{x} + x^-$ to the ideal functionality.

We show that the joint distribution of the view of the adversary and the output of the honest party is distributed identically in the hybrid and the ideal executions. This is done easily in the case where $n/2-1$ or more parties abort at some round $r > r^*$ (and thus, both parties receive the correct output $f_{t,n-t}(x,y)$). Now, we consider the case where $r \leq r^*$. The view of the adversary holding input $\boldsymbol{x}$ in the r-th round consists of: $a^{r,R',n',z}$ for all (R',n',z) such that $a_{(R',n',z)} \neq \perp$.

The view of the adversary until round i is distributed identically in both executions. Thus, all that is left to show is that the view of the adversary in the last round and the output of the honest party are distributed identically in both executions. That is, we show that for every $(\boldsymbol{a}, b)$, where $b \in \{0,1\}$ and $\boldsymbol{a}$ is such that all indeces in $\phi(\boldsymbol{x})$ are set to 0/1 and all other indeces are set to $\perp$, it is the case that:

$$\Pr[(\mathsf{View}^r_{\mathsf{hyb}}, \mathsf{Out}_{\mathsf{hyb}}) = (\boldsymbol{a}, b) \mid r \leq r^*] = \Pr[(\mathsf{View}^r_{\mathsf{ideal}}, \mathsf{Out}_{\mathsf{ideal}}) = (\boldsymbol{a}, b) \mid r \leq r^*]. \tag{4.1}$$

Formally, $(\mathsf{View}^r_{\mathsf{hyb}}, \mathsf{Out}_{\mathsf{hyb}})$ and $(\mathsf{View}^r_{\mathsf{ideal}}, \mathsf{Out}_{\mathsf{ideal}})$ denote the entire view and output in the hybrid and ideal execution. Note that $\mathsf{View}^r_{\mathsf{hyb}}, \mathsf{View}^r_{\mathsf{ideal}}$ actually

consist of secret shares, whereas $\boldsymbol{a}$ denotes the reconstructed values for the instances that can be opened by the adversary. We simplify our computations by assuming that the views $\mathsf{View}^r_{\mathsf{hyb}}, \mathsf{View}^r_{\mathsf{ideal}}$ consist only of the values the adversary can *reconstruct* given its set of shares, and not the shares themselves. Given the "perfect privacy" property of sharing schemes (see [5]), if the probabilities are the same with respect to the reconstructed values, then they will also be the same with respect to the original view.

Implicit in our argument, is that—in the hybrid execution—the output b of the honest parties in round $r < \mathsf{rounds}$ is independent of the view of the adversary, represented by $\boldsymbol{a}$. While this is trivially true in the two-party case, It is not as obvious in our protocol, since when $n/2$ or $n/2+1$ parties are corrupted, the adversary can open many instances of the secret sharing scheme. Specifically, we must show that for the instance used in Step 4 to reconstruct—identified by $(\overline{R}', n-t', *)$—it is always the case that $a_{(\overline{R}', n-t', *)} = \perp$.

This will follow from the following property that is straightforward to check:

Property 1. Let t be the number of corruptions. If for some $(R_1, n_1, z_1), (R_2, n_2, z_2)$, $a_{(R_1,n_1,z_1)} \neq \perp$ and $a_{(R_2,n_2,z_2)} \neq \perp$ then

$$|R_1 \cup R_2| + \max(n_1 - |R_1|, n_2 - |R_2|) \leq t.$$

Recall that the set of corrupted parties is denoted by T, and the set of parties who abort and/or do not submit valid input in Step 4 is denoted T'. Let $R' = T' \cap R$. Let $|T'| = t'$. Then parties reconstruct with $S' := \overline{T}'$, $n' = n - t'$, and $\overline{R}' = \overline{T}' \cap R$. Note that $\{R', \overline{R}'\}$ form a partition of R. Note that corrupted parties can open $(R', t', *)$, while the parties in S' can open $(\overline{R}', n-t', *)$. Assume towards contradiction that the adversary can also open $(\overline{R}', n-t', *)$. Note that $(t'-|R'|)+(n-t'-|\overline{R}'|) = n-|R| = n-3$. Therefore, $\max(t'-|R'|, n-t'-|\overline{R}'|) \geq n/2-1$. Thus,

$$|R' \cup \overline{R}'| + \max(t' - |R'|, n - t' - |\overline{R}'|) \geq 3 + n/2 - 1 > n/2 + 1 \geq t,$$

which contradicts Property 1.

We now show that Eq. (4.1) holds by considering all possible values for $(\boldsymbol{a}, b)$. First, observe that

$$\Pr[r = r^* \mid r \leq r^*] = \alpha \text{ and } \Pr[r < i^* \mid r \leq i^*] = 1 - \alpha.$$

In the following we will consider only valid parameter settings $(T, t, \boldsymbol{x}, T', t', x^+, x^-, \boldsymbol{a})$.

In case $f^{x^-}_{t',n-t}(x^+, y) = 0$. Let $\boldsymbol{a}' \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$. Let $S_{\boldsymbol{a}'}$ be the set of positions in $\boldsymbol{a}'$ that are set to 0 and $S'_{\boldsymbol{a}'}$ be the set of positions in $\boldsymbol{a}'$ that are set to 1. $\boldsymbol{a}_0, \boldsymbol{a}_1$ are defined as before. For condensed notation, we let $\boldsymbol{a}_0, \boldsymbol{a}_1, \boldsymbol{a}'$ be indexed by $\mathsf{t} = (R', n', z)$.

View	Real	Ideal
$(\boldsymbol{a}_0, 0)$	$\alpha \cdot (1 - p_y^{x^-,t,t'}) + (1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot (1 - p_y^{x^-,t,t'})$	$\alpha + (1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot (1 - q_y^{x^+,x^-,\boldsymbol{a}_0})$
$(\boldsymbol{a}_0, 1)$	$\alpha \cdot p_y^{x^-,t,t'} + (1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot p_y^{x^-,t,t'}$	$(1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot (q_y^{x^+,x^-,\boldsymbol{a}_0})$
$(\boldsymbol{a}', 0)$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (1 - p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (1 - q_y^{x^+,x^-,\boldsymbol{a}'})$
$(\boldsymbol{a}', 1)$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (q_y^{x^+,x^-,\boldsymbol{a}'})$
$(\boldsymbol{a}_1, 0)$	$(1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (1 - p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (1 - q_y^{x^+,x^-,\boldsymbol{a}_1})$
$(\boldsymbol{a}_1, 1)$	$(1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (q_y^{x^+,x^-,\boldsymbol{a}_1})$

In the table, we compute the probabilities of representative choices of $(\boldsymbol{a}, b)$ in the Real and Ideal worlds:

It can be seen that for $\boldsymbol{a}' \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$ we get the following constraint: $q_y^{x^+,x^-,\boldsymbol{a}'} = p_y^{x^-,t,t'}$. Thus, all constraints for $\boldsymbol{a}' \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$ can be satisfied by setting $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} = \boldsymbol{X}_{real,t'}$.

Additionally, we obtain the constraints:

$$q_y^{x^+,x^-,\boldsymbol{a}_0} = p_y^{x^-,t,t'} + \frac{\alpha \cdot p_y^{x^-,t,t'}}{(1-\alpha) \cdot \prod_{(R',n',z)}(\overline{p}_x^{(R',n',z)})})$$

and

$$q_y^{x^+,x^-,\boldsymbol{a}_1} = p_y^{x^-,t,t'},$$

which are satisfied according to our assumptions in the theorem.

In case $f_{t',n-t}^{x^-}(x^+, y) = 1$. Let $\boldsymbol{a}' \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$. Let $S_{\boldsymbol{a}'}$ be the set of positions in $\boldsymbol{a}'$ that are set to 0 and $S'_{\boldsymbol{a}'}$ be the set of positions in $\boldsymbol{a}'$ that are set to 1. $\boldsymbol{a}_0, \boldsymbol{a}_1$ are defined as before. For condensed notation, we let $\boldsymbol{a}_0, \boldsymbol{a}_1, \boldsymbol{a}'$ be indexed by $\mathsf{t} = (R', n', z)$.

In the table, we compute the probabilities of representative choices of $(\boldsymbol{a}, b)$ in the Real and Ideal worlds:

View	Real	Ideal
$(\boldsymbol{a}_0, 0)$	$(1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot (1 - p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot (1 - q_y^{x^+,x^-,\boldsymbol{a}_0})$
$(\boldsymbol{a}_0, 1)$	$(1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}}) \cdot p_y^{x^-,t,t'}$	$(1-\alpha)\prod_{\mathsf{t}}(\overline{p}_x^{\mathsf{t}})(q_y^{x^+,x^-,\boldsymbol{a}_0})$
$(\boldsymbol{a}', 0)$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (1 - p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (1 - q_y^{x^+,x^-,\boldsymbol{a}'})$
$(\boldsymbol{a}', 1)$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}\in S_{\boldsymbol{a}'}}(\overline{p}_x^{\mathsf{t}})\prod_{\mathsf{t}\in S'_{\boldsymbol{a}'}}(p_x^{\mathsf{t}}) \cdot (q_y^{x^+,x^-,\boldsymbol{a}'})$
$(\boldsymbol{a}_1, 0)$	$\alpha \cdot (1 - p_y^{x^-,t,t'}) + (1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (1 - p_y^{x^-,t,t'})$	$(1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (1 - q_y^{x^+,x^-,\boldsymbol{a}_1})$
$(\boldsymbol{a}_1, 1)$	$\alpha \cdot p_y^{x^-,t,t'} + (1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (p_y^{x^-,t,t'})$	$\alpha + (1-\alpha)\prod_{\mathsf{t}}(p_x^{\mathsf{t}}) \cdot (q_y^{x^+,x^-,\boldsymbol{a}_1})$

It can be seen that for $\boldsymbol{a}' \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$ we get the following constraint: $q_y^{x^+,x^-,\boldsymbol{a}'} = p_y^{x^-,t,t'}$. Thus, all constraints for $\boldsymbol{a}' \notin \{\boldsymbol{a}_0, \boldsymbol{a}_1\}$ can be satisfied by setting $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} = \boldsymbol{X}_{real,t'}$.

Additionally, we obtain the constraints:

$$q_y^{x^+,x^-,\boldsymbol{a}_1} = p_y^{x^-,t,t'} + \frac{\alpha \cdot (p_y^{x^-,t,t'} - 1)}{(1-\alpha) \cdot \prod_{(R',n',z)} p_x^{(R',n',z)}})$$

and

$$q_y^{x^+,x^-,\boldsymbol{a}_0} = p_y^{x^-,t,t'}.$$

Since $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} \cdot \boldsymbol{M}_{f_{t',n-t}^{x^-}} = \boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}}$, the above constraints are satisfied.

This concludes the proof of Theorem 6.

The following lemma concludes the analysis of the protocol in Fig. 3:

Lemma 1. *There exists $\alpha = 1/\operatorname{poly}(n)$ such that for every valid setting of parameters $(T, t, \boldsymbol{x}, T', t', x^+, x^-, \boldsymbol{a})$, there exists a probability vector $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}}$ such that $\boldsymbol{X}_{ideal,t'}^{x^+,x^-,\boldsymbol{a}} \cdot \boldsymbol{M}_{f_{t',n-t}^{x^-}} = \boldsymbol{Q}^{x^+,x^-,\boldsymbol{a}}$.*

Proof. We begin by proving the lemma for the special case where $t = n/2 + 1$, $t' = n/2 - 1$ and $x^- = 1$.

Define $\boldsymbol{P}_y^{1,n/2+1,n/2-1} = (p_{y=n/2-1}^{1,n/2+1,n/2-1}, \ldots, p_{y=0}^{1,n/2+1,n/2-1})$.

Note that the output of the function $f_{n/2-1,n/2-1}^1$ is 1 in position $[x, y]$ if the sum of $x + y \geq n/2$. In the following example we set $n/2 - 1 = 3$. The truthtable of $f_{n/2-1,n/2-1}^1$ is as follows:

	$y=3$	$y=2$	$y=1$	$y=0$
$x=0$	0	0	0	0
$x=1$	1	0	0	0
$x=2$	1	1	0	0
$x=3$	1	1	1	0

And becomes the following in matrix form:

$$\boldsymbol{M}_{f_{n/2-1,n/2-1}^1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Since the final column of the matrix is all 0, we can simply remove it, since $p_{y=0}^{1,n/2+1,n/2-1} = 0$, $q_{y=0}^{x^+,1,\boldsymbol{a}_0} = 0$, and $q_{y=0}^{x^+,1,\boldsymbol{a}_1} = 0$. Thus, $\boldsymbol{M}_{f_{n/2-1,n/2-1}^1}$ denotes the above matrix with the final column deleted.

For every valid $(T, t = n/2 + 1, \boldsymbol{x}, T', t' = n/2 - 1, x^+, x^- = 1, \boldsymbol{a})$ we need to find a vector $\boldsymbol{s} \in \mathbb{R}^{n/2-1}$ such that $\boldsymbol{s}\boldsymbol{M}_{f_{n/2-1,n/2-1}^1} = \boldsymbol{Q}^{x^+,1,\boldsymbol{a}}$ and the vector

$\boldsymbol{s} = s_0, \ldots, s_{n/2-1}$ further needs to correspond to a probability distribution– i.e. we require that $\sum_{k=0}^{n/2-1} s_k = 1$. In addition, we require that each s_k is non-negative.

Let $\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}}$ denote the matrix obtained when a column vector of 1's is concatenated with the matrix $\boldsymbol{M}_{f^1_{n/2-1,n/2-1}}$. For the case $n/2-1=3$, we obtain the following:

$$\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}} = \begin{bmatrix} 1\,0\,0\,0 \\ 1\,1\,0\,0 \\ 1\,1\,1\,0 \\ 1\,1\,1\,1 \end{bmatrix}$$

We need to find a setting of α, such that $\alpha = 1/\operatorname{poly}(n)$ and such that the unique solution for $\boldsymbol{s}$, where $\boldsymbol{s}\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}} = (1||\boldsymbol{Q}^{x,1,0,\alpha})$ is non-negative. In the following, we argue that by setting α sufficiently small, but still $1/\operatorname{poly}(n)$ (yielding a protocol with $1/\alpha \cdot \omega(\log(\lambda)) = \operatorname{poly}(n,\lambda)$ rounds), we can find such a solution.

We know there is a non-negative solution $\boldsymbol{s}$ to $\boldsymbol{s}\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}} = (1||\boldsymbol{P}_y^{1,n/2+1,n/2-1})$. In fact, the solution is simply $\boldsymbol{s} = (\frac{1}{n/2}, \ldots, \frac{1}{n/2})$, as this is the distribution $\boldsymbol{X}_{real,t'=n/2-1}$ over inputs $\hat{x} \in \{0, \ldots, n/2-1\}$ that produces the real output distribution $\boldsymbol{P}_y^{1,n/2+1,n/2-1}$. Note that $\boldsymbol{s}$ has distance at least $2/n$ from any vector with negative entries (since each coordinate of $\boldsymbol{s}$ has magnitude $2/n$). If $(1||\boldsymbol{Q}^{x-1,1,\boldsymbol{a}}) = (1||\boldsymbol{P}_y^{1,n/2+1,n/2-1}) + \boldsymbol{w}$, where $\boldsymbol{w}$ is a vector with magnitude at most d, we have that

$$(\boldsymbol{s}+\boldsymbol{s}')\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}} = \boldsymbol{s}\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}} + \boldsymbol{s}'\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}} = (1||\boldsymbol{P}_y^{1,n/2+1,n/2-1}) + \boldsymbol{w},$$

where

$$\boldsymbol{s}' = \boldsymbol{w}(\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}})^{-1}.$$

Now, the matrix $(\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}})^{-1}$ has the following form:

$$(\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}})^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

In other words, the diagonal entries are set to 1, the second diagonal entries are set to -1 and all other entries are set to 0. We upper bound the spectral norm of $(\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}})^{-1}$ by $\sqrt{5}$ (see full version). Bounding the spectral norm of $(\boldsymbol{M}^{+}_{f^1_{n/2-1,n/2-1}})^{-1}$ by $\sqrt{5}$ guarantees that since $\boldsymbol{w}$ has magnitude d, $\boldsymbol{s}'$ has magnitude at most $d' = \sqrt{5} \cdot d$. By choosing $d = \frac{2}{\sqrt{5} \cdot n}$, we have that $(\boldsymbol{s}+\boldsymbol{s}')$ has all non-negative entries. To ensure that $\boldsymbol{w}$ has magnitude at most d, it is

sufficient to ensure that each coordinate of $\boldsymbol{w} = (1||\boldsymbol{Q}^{x-1,1,\boldsymbol{a}}) - (1||\boldsymbol{P}_y^1)$ has magnitude at most $d/\sqrt{n}$. This can be achieved by setting $\alpha \leq 1/2$ such that

$$\frac{2\alpha}{\prod_{(R',n',z)} p_x^{(R',n',z)}} \leq d/\sqrt{n} \quad \text{and} \quad \frac{2\alpha}{\prod_{(R',n',z)} (\overline{p}_x^{(R',n',z)})} \leq d/\sqrt{n}. \tag{4.2}$$

Now, both $p_x^{(R',n',z)}$ and $\overline{p}_x^{(R',n',z)}$ must be at least $1/(n/2+2)$, since if they are not identically 0 (resp. identically 1), then there is at least one value of $\hat{y} \in \{0,\ldots,n-n'\}$ for which $f_{n',n-n'}(z,\hat{y}) = 1$ (resp. $f_{n',n-n'}(z,\hat{y}) = 0$) and since $n' \geq n/2-1$, $\Pr_{\hat{y}\sim\{0,\ldots,n-n'\}}[f_{n',n-n'}(z,\hat{y}) = 1] \geq 1/(n/2+2) > 1/n$ (resp. $\Pr_{\hat{y}\sim\{0,\ldots,n-n'\}}[f_{n',n-n'}(z,\hat{y}) = 0] \geq 1/(n/2+2) > 1/n$). Since, furthermore, $|\phi(\boldsymbol{x})| \leq \mathsf{deg}$, $\prod_{(R',n',z)} p_x^{(R',n',z)} \geq 1/n^{\mathsf{deg}}$ and $\prod_{(R',n',z)} (\overline{p}_x^{(R',n',z)}) \geq 1/n^{\mathsf{deg}}$. Thus, (4.2) is achieved by setting $\alpha \leq \frac{d}{2n^{\mathsf{deg}+0.5}}$. Finally, plugging in $d = \frac{2}{\sqrt{5}\cdot n}$, we have that $\alpha \leq \frac{1}{\sqrt{5}n^{\mathsf{deg}+1.5}}$. This results in a number of rounds $\omega(\log(\lambda)) \cdot 1/\alpha$, which is polynomial in the security parameter λ and in the number of parties n.

We now formalize the argument for any setting of $t = n/2+1$, $t' = n/2-1$ and $x^- = 1$. In fact, we see that the only thing that changes in the argument is $\boldsymbol{M}^+_{f^1_{n/2-1,n/2-1}}$. We must prove that $\boldsymbol{M}^+_{f^{x^-}_{t',n-t}}$ is invertible and that the spectral norm of $(\boldsymbol{M}^+_{f^{x^-}_{t',n-t}})^{-1}$ is bounded by $\sqrt{5}$.

In fact, we will show something slightly more general: For any m,n and any threshold th, consider the function $f^{th}_{m,n} : \{0,\ldots,m\} \times \{0,\ldots,n\}$ defined as: $f^{th}_{m,n}(x,y) = 1$ iff $x+y \geq th$. For non-triviality, we assume that $th > 0$ and that $m+n \geq th$. Consider the matrix $\boldsymbol{M}_{f^{th}_{m,n}}$.

We begin by removing from $\boldsymbol{M}_{f^{th}_{m,n}}$ columns that are all 0. I.e. columns $y = k$ such that $m+k < th$. The number of columns removed is $\ell_0 := th - m$, if $th - m \geq 1$ and 0 otherwise.

We next remove from $\boldsymbol{M}_{f^{th}_{m,n}}$ any columns ($y = k$) that are all 1 (this is ok since in this case $p^{x^+,x^-}_{y=k} = 1$, $q^{x^+,x^-,\boldsymbol{a}_0}_{y=k} = 1$, and $q^{x^+,x^-,\boldsymbol{a}_1}_{y=k} = 1$, and since the column will be added back at the end). Column $y = k$ will be all 1 if $k \geq th$. The number of columns removed is $\ell_1 := n - th + 1$, if $n - th + 1 \geq 1$ and 0 otherwise.

Now, we will show that the number of columns remaining ($(n+1) - \ell_1 - \ell_0$) is at least one fewer than the number of rows ($m+1$). The number of columns remaining is

$$\begin{aligned}(n+1) - \ell_1 - \ell_0 &\leq (n+1) - (th-m) - (n-th+1) \\ &= n+1-th+m-n+th-1 = m.\end{aligned}$$

Furthermore, if $m+1 > (n+1) - \ell_1 - \ell_0 + 1$, then there must be two identical rows, one of which can be removed. Therefore, after removing the columns, removing duplicate rows and adding a column of 1's, $\boldsymbol{M}^+_{f^{th}_{m,n}}$ has the form of a (non-singular) lower triangular matrix with 1's in each lower triangular entry and dimension $(n+2-\ell_1-\ell_0) \times (n+2-\ell_1-\ell_0)$.

4.1 Implementing the Dealer and $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$

Implementing the Dealer proceeds almost the same as the case of $n/2$ or fewer corruptions described in Sect. 3.1. The only differences are that we use a different access structure for the inner/outer secret-sharing schemes. Specifically, in round r, party P_j learns a share in a secret sharing scheme for access structure $\mathcal{A}_{R,R',n'-z,2n}$, for every $R' \subseteq R$, $n' \in [n], z \in \{0,\ldots,n'\}$ (we call these P_j's shares of the inner secret-sharing scheme).

For each round, the shares of each party P_j are then shared in a *special* 2-out-of-2 secret-sharing scheme, where P_j gets one of the two shares (called the mask). In addition, all parties (including P_j) receive shares in a $n/2+2$-out-of-n Shamir secret-sharing scheme of the other share of the 2-out-of-2 secret sharing. We call the resulting secret-sharing scheme the *outer* $(n/2+2)$-out-of-n scheme (since $n/2+1$ parties and the holder of the mask are needed to reconstruct the secret).

To implement ideal functionality $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$, when $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$ is run in Step 4, not all parties remaining in the sets S and S' are necessarily honest. However, our restriction on $n \geq 8$ ensures that S and S' contains an honest majority. Therefore, $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$ can be implemented with a fully secure protocol (with fairness and guaranteed output delivery). When $F^{\mathsf{th},n/2+1}_{\mathsf{Recon},S,p}$ is run in Step 5, there may *not* be an honest majority, and the same approach from the previous section (Sect. 3.1) works.

References

1. Alon, B., Omri, E.: Almost-optimally fair multiparty coin-tossing with nearly three-quarters malicious. In: Hirt, M., Smith, A. (eds.) TCC 2016, Part I. LNCS, vol. 9985, pp. 307–335. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53641-4_13
2. Asharov, G.: Towards characterizing complete fairness in secure two-party computation. In: Lindell [32], pp. 291–316
3. Asharov, G., Beimel, A., Makriyannis, N., Omri, E.: Complete characterization of fairness in secure two-party computation of boolean functions. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 199–228. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46494-6_10
4. Asharov, G., Lindell, Y., Rabin, T.: A full characterization of functions that imply fair coin tossing and ramifications to fairness. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 243–262. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_14
5. Beimel, A.: Secret-sharing schemes: a survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 11–46. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20901-7_2
6. Beimel, A., Haitner, I., Makriyannis, N., Omri, E.: Tighter bounds on multi-party coin flipping via augmented weak martingales and differentially private sampling. In: Thorup, M. (ed.) 59th FOCS, pp. 838–849. IEEE Computer Society Press, October 2018

7. Beimel, A., Lindell, Y., Omri, E., Orlov, I.: $1/p$-secure multiparty computation without honest majority and the best of both worlds. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 277–296. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_16
8. Beimel, A., Omri, E., Orlov, I.: Protocols for multiparty coin toss with dishonest majority. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 538–557. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_29
9. Beimel, A., Omri, E., Orlov, I.: Secure multiparty computation with partial fairness. Cryptology ePrint Archive, Report 2010/599 (2010). http://eprint.iacr.org/2010/599
10. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: STOC 1988 [39], pp. 1–10
11. Buchbinder, N., Haitner, I., Levi, N., Tsfadia, E.: Fair coin flipping: tighter analysis and the many-party case. In: Klein, P.N. (ed.) 28th SODA, pp. 2580–2600. ACM-SIAM, January 2017
12. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: STOC 1988 [39], pp. 11–19
13. Cleve, R.: Limits on the security of coin flips when half the processors are faulty (extended abstract). In: 18th ACM STOC, pp. 364–369. ACM Press, May 1986
14. Cohen, R., Lindell, Y.: Fairness versus guaranteed output delivery in secure multiparty computation. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 466–485. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_25
15. Dachman-Soled, D., Lindell, Y., Mahmoody, M., Malkin, T.: On the black-box complexity of optimally-fair coin tossing. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 450–467. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_27
16. Dachman-Soled, D., Mahmoody, M., Malkin, T.: Can optimally-fair coin tossing be based on one-way functions? In: Lindell [32], pp. 217–239
17. Goldreich, O.: The Foundations of Cryptography - Volume 2: Basic Applications. Cambridge University Press, Cambridge (2004)
18. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC, pp. 218–229. ACM Press, May 1987
19. Gordon, S.D., Hazay, C., Katz, J., Lindell, Y.: Complete fairness in secure two-party computation. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 413–422. ACM Press, May 2008
20. Gordon, D., Ishai, Y., Moran, T., Ostrovsky, R., Sahai, A.: On complete primitives for fairness. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 91–108. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_7
21. Gordon, S.D., Katz, J.: Complete fairness in multi-party computation without an honest majority. In: Reingold [38], pp. 19–35
22. Gordon, S.D., Katz, J.: Partial fairness in secure two-party computation. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 157–176. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_8
23. Dov Gordon, S., Liu, F.-H., Shi, E.: Constant-round MPC with fairness and guarantee of output delivery. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II. LNCS, vol. 9216, pp. 63–82. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_4

24. Haitner, I., Makriyannis, N., Omri, E.: On the complexity of fair coin flipping. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part I. LNCS, vol. 11239, pp. 539–562. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03807-6_20
25. Haitner, I., Tsfadia, E.: An almost-optimally fair three-party coin-flipping protocol. In: Shmoys, D.B. (ed.) 46th ACM STOC, pp. 408–416. ACM Press, May/June 2014
26. Hirt, M., Maurer, U.M.: Complete characterization of adversaries tolerable in secure multi-party computation (extended abstract). In: Burns, J.E., Attiya, H. (eds.) 16th ACM PODC, pp. 25–34. ACM, August 1997
27. Hirt, M., Maurer, U.M.: Player simulation and general adversary structures in perfect multiparty computation. J. Cryptol. **13**(1), 31–60 (2000)
28. Ishai, Y., Katz, J., Kushilevitz, E., Lindell, Y., Petrank, E.: On achieving the best of both worlds in secure multiparty computation. Cryptology ePrint Archive, Report 2010/029 (2010). http://eprint.iacr.org/2010/029
29. Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: On combining privacy with guaranteed output delivery in secure multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 483–500. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_29
30. Ishai, Y., Ostrovsky, R., Seyalioglu, H.: Identifying cheaters without an honest majority. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 21–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28914-9_2
31. Ishai, Y., Ostrovsky, R., Zikas, V.: Secure multi-party computation with identifiable abort. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 369–386. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_21
32. Lindell, Y. (ed.): TCC 2014. LNCS, vol. 8349. Springer, Heidelberg (2014)
33. Lindell, Y., Rabin, T.: Secure two-party computation with fairness - a necessary design principle. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 565–580. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_19
34. Moran, T., Naor, M., Segev, G.: An optimally fair coin toss. In: Reingold [38], pp. 1–18
35. Moran, T., Naor, M., Segev, G.: An optimally fair coin toss. J. Cryptol. **29**(3), 491–513 (2016)
36. O'Donnell, R.: Analysis of Boolean Functions. Cambridge University Press, Cambridge (2014)
37. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: 21st ACM STOC, pp. 73–85. ACM Press, May 1989
38. Reingold, O. (ed.): TCC 2009. LNCS, vol. 5444. Springer, Heidelberg (2009)
39. 20th ACM STOC. ACM Press, May 1988