Passive Corruption in Statistical Multi-Party Computation

(Extended Abstract)*

Martin Hirt[1], Christoph Lucas[1], Ueli Maurer[1], and Dominik Raub[2]

[1] Department of Computer Science, ETH Zurich, Switzerland
{hirt,clucas,maurer}@inf.ethz.ch
[2] Department of Computer Science, University of Århus, Denmark
raub@cs.au.dk

Abstract. The goal of *Multi-Party Computation* (MPC) is to perform an arbitrary computation in a distributed, private, and fault-tolerant way. For this purpose, a fixed set of n parties runs a protocol that tolerates an adversary corrupting a subset of the parties, preserving certain security guarantees like correctness, secrecy, robustness, and fairness. Corruptions can be either *passive* or *active*: A passively corrupted party follows the protocol correctly, but the adversary learns the entire internal state of this party. An actively corrupted party is completely controlled by the adversary, and may deviate arbitrarily from the protocol. A *mixed adversary* may at the same time corrupt some parties actively and some additional parties passively.

In this work, we consider the statistical setting with mixed adversaries and study the exact consequences of active and passive corruptions on secrecy, correctness, robustness, and fairness separately (i.e., hybrid security). Clearly, the number of passive corruptions affects the thresholds for secrecy, while the number of active corruptions affects all thresholds. It turns out that in the statistical setting, the number of passive corruptions in particular also affects the threshold for correctness, i.e., in all protocols there are (tolerated) adversaries for which a single additional passive corruption is sufficient to break correctness. This is in contrast to both the perfect and the computational setting, where such an influence cannot be observed. Apparently, this effect arises from the use of information-theoretic signatures, which are part of most (if not all) statistical protocols.

Keywords: Multi-party computation, passive corruption, statistical security, hybrid security, mixed adversaries.

* The full version of this paper is available at the *Cryptology ePrint Archive*: http://eprint.iacr.org/2012/272. This work was partially supported by the Zurich Information Security Center.

A. Smith (Ed.): ICITS 2012, LNCS 7412, pp. 129–146, 2012.

1 Introduction

1.1 Secure Multi-Party Computation

Multi-Party Computation (MPC) allows a set of n parties to securely perform an arbitrary computation in a distributed manner, where security means that secrecy of the inputs and correctness of the output are maintained even when some of the parties are dishonest. The dishonesty of parties is modeled with a central adversary who corrupts parties. The adversary can be *passive*, i.e. can read the internal state of the corrupted parties, or *active*, i.e., can make the corrupted parties deviate arbitrarily from the protocol.

MPC was originally proposed by Yao [Yao82]. The first general solution was provided in [GMW87], where, based on computational intractability assumptions, security against a passive adversary was achieved for $t < n$ corruptions, and security against an active adversary was achieved for $t < \frac{n}{2}$. Information-theoretic security was achieved in [BGW88, CCD88] at the price of lower corruption thresholds, namely $t < \frac{n}{2}$ for passive and $t < \frac{n}{3}$ for active adversaries. The latter bound can be improved to $t < \frac{n}{2}$ if both broadcast channels are assumed and a small error probability is tolerated [RB89, Bea89]. These results were generalized to the non-threshold setting, where the corruption capability of the adversary is not specified by a threshold t, but rather by a so called adversary structure $\mathcal{Z}$, a monotone collection of subsets of the player set, where the adversary can corrupt the players in one of these subsets [HM97].

All mentioned protocols achieve full security, i.e. secrecy, correctness, and robustness. *Secrecy* means that the adversary learns nothing about the honest parties' inputs and outputs (except, of course, for what can be derived from the corrupted parties' inputs and outputs). *Correctness* means that all parties either output the right value or no value at all. *Robustness* means that the adversary cannot prevent the honest parties from learning their respective outputs. This last requirement turns out to be very demanding. Therefore, relaxations of full security have been proposed, where robustness is replaced by weaker output guarantees: *Fairness* means that the adversary can possibly prevent the honest parties from learning their outputs, but then also the corrupted parties do not learn their outputs. *Agreement on abort* means that the adversary can possibly prevent honest parties from learning their output, even while corrupted parties learn their outputs, but then the honest parties at least reach agreement on this fact (and typically make no output). In our constructions, all abort decisions are based on publicly known values. Hence, we have agreement on abort for free.[1]

The traditional setting of MPC has been generalized in two directions. On the one hand, the notion of *hybrid security* was introduced to allow for protocols with different security guarantees depending on the number of corruptions [Cha89, FHHW03, FHW04, IKLP06, Kat07, LRM10, HLMR11]. Intuitively, the more corrupted parties, the less security is guaranteed. This model also allows to analyze each security guarantee separately and independent of other guarantees. On the other hand, protocols were presented that do not restrict the adversary to

[1] The impossibility proof holds even when agreement on abort is not required.

a single corruption type [Cha89, DDWY93, FHM98, FHM99, BFH+08, HMZ08, HLMR11]. The *mixed adversaries* considered there can perform each corruption with one out of several corruption types. This allows to consider e.g. active and passive corruption in the same protocol execution.

1.2 Contributions

In this work, we consider a setting with mixed adversaries and hybrid security. This allows, for the first time, to separately analyze the relation between passive corruption and the various security guarantees. It turns out that, in the statistical model, passive corruption does not only affect secrecy, but in particular also correctness. In most statistically secure protocols, some kind of information-theoretic signature is used. When combining active and passive corruptions, one inherent problem of any kind of information-theoretic signature is that passively corrupted parties cannot reliably verify signed values. Existing protocols for the statistical setting assume an honest majority. Therefore, a simple majority vote on the signature guarantees reliable verification even for passively corrupted parties. In this work, we show that this assumption is too strong, and that signatures can be used even without an honest majority. As the main technical contribution, we provide optimal protocols for both general and threshold adversaries that cope with this issue. As a new technique for the setting with general adversaries, we introduce *group commitments*, a non-trivial extension of IC-Signatures, which might be of independent interest.

Furthermore, we introduce the notion of *multi-thresholds*. To the best of our knowledge, all known protocols for threshold mixed adversaries (e.g. [FHM98]) characterize the tolerable adversaries with a single pair of thresholds (one threshold for the number of actively, and one for the number of passively corrupted parties). This pair represents the single maximal adversary that can be tolerated. We generalize this basic characterization to allow for several incomparable maximal adversaries. It turns out that, in our setting, multi-thresholds allow to construct protocols that tolerate strictly more adversaries than a single pair of thresholds, without losing efficiency.

1.3 Model

We consider n parties $p_1, \ldots, p_n$, connected by pairwise synchronous secure channels and authenticated broadcast channels[2], who want to compute some probabilistic function over a finite field $\mathbb{F}$, represented as circuit with input, addition, multiplication, random, and output gates. This function can be reactive, where parties can provide further inputs after having received some intermediate outputs.

There is a central adversary with unlimited computing power who corrupts some parties passively (and reads their internal state) or even actively (and

[2] In [PW92] it is shown how broadcast can be implemented given a setup.

makes them misbehave arbitrarily). We denote the actual sets of actively (passively) corrupted parties by $\mathcal{D}^*$ ($\mathcal{E}^*$), where $\mathcal{D}^* \subseteq \mathcal{E}^*$. Uncorrupted parties are called *honest*, non-actively corrupted parties are called *correct*. The security of our protocols is statistical, i.e. information-theoretic with a small error probability. We say a security guarantee holds *statistically* if it holds with overwhelming probability. The guaranteed security properties (secrecy, correctness, fairness, robustness, agreement on abort) depend on $(\mathcal{D}^*, \mathcal{E}^*)$.

For ease of notation, we assume that if a party does not receive an expected message (or receives an invalid message), a default message is used instead. Furthermore, we use subprotocols that might abort. Such an abort is always global, i.e., if any subprotocol aborts, the whole protocol execution halts.

In the analysis of our protocols, we assume "instant randomness", i.e. parties generate their randomness on the fly when needed in the protocol run. This allows even passively corrupted parties to e.g. choose challenges in zero-knowledge proofs that are unpredictable to the adversary. Note that in a setting without secrecy, we have no input independence[3]. Hence, standard techniques (e.g. Blum coin-toss) to jointly generate these challenges are insecure.

1.4 Outline of the Paper

The paper is organized as follows: In Sec. 2, we present information checking, which is used as a basic primitive in our protocols. As a main technical contribution, in Sections 3 and 4, we present protocols for the model with mixed adversaries and hybrid security for both general and threshold adversaries, together with optimal bounds. In Sec. 5, we provide conclusions of our results.

2 Information Checking

Information checking (IC) [RB89, CDD+99] is a primitive that allows a sender to send a value to an intermediary, such that when the receiver obtains this value from the intermediary, he can check that this is indeed the value from the sender. When all parties act as receivers, this primitive is called *IC signature*, and the sender is called signer. IC signatures are realized using a pair of protocols IC-Sign and IC-Reveal. IC-Sign allows a signer to sign a value for a particular intermediary (while providing secrecy with respect to the remaining parties), and IC-Reveal allows this intermediary to verifiably forward this value to all other parties.

More precisely, let $\langle v \rangle_{i,j}$ denote the state of all players where a value v is *IC-signed* (or simply *signed*) by signer p_i for intermediary p_j. In analogy to traditional signatures, we equivalently say that the intermediary p_j *holds* the signature $\langle v \rangle_{i,j}$. We require that a default signature $\langle v \rangle_{i,j}$ can always be generated given that all parties know the value v, and that signatures are linear, i.e., the

[3] That means, the adversary can choose the inputs of actively corrupted parties after learning the inputs of correct parties.

sum of two signatures $\langle v \rangle_{i,j}$ and $\langle v' \rangle_{i,j}$ from signer p_i to intermediary p_j for values v and v', respectively, is a signature from p_i to p_j for the sum $v+v'$. IC-Sign is a protocol that, given a signer p_i and an intermediary p_j that both know the same value v, provides the following guarantees: If p_i and p_j are correct, IC-Sign correctly computes a valid signature $\langle v \rangle_{i,j}$ on v without leaking any information about v to the remaining parties. Otherwise, IC-Sign either correctly computes a valid signature $\langle v \rangle_{i,j}$ on v, or all (correct) parties output $\perp$, with overwhelming probability. Given a signature $\langle v \rangle_{i,j}$, IC-Reveal robustly computes the output $x_k \in \{(\text{“accept”}, v'), \text{“reject”}\}$ for each p_k. We make the following correctness requirements: If p_j is correct, all correct parties p_k output $x_k = (\text{“accept”}, v)$. Else, if both p_i and p_k are honest, then $x_k \in \{(\text{“accept”}, v), \text{“reject”}\}$ (with overwhelming probability, even when p_j is active). Note that we do not require agreement on the output of correct parties in IC-Reveal. Furthermore, if p_j is active and p_i or p_k is not honest, then p_k might output $x_k = (\text{“accept”}, v')$ for $v' \neq v$.

In the full version of this paper, we provide an instantiation of IC signatures.

3 MPC with General Adversaries

Traditionally, protocols for general adversaries are characterized by an adversary structure $\mathcal{Z}$ that specifies the tolerated subsets of the player set [HM97]. For our setting, we have to extend this basic representation: On the one hand, we consider mixed adversaries, which are characterized by adversary structures consisting of tuples $(\mathcal{D}, \mathcal{E})$ of subsets of $\mathcal{P}$, where the adversary may corrupt the parties in $\mathcal{E}$ passively, and the parties in $\mathcal{D} \subseteq \mathcal{E}$ even actively. On the other hand, each security guarantee depends on the sets of *actually* corrupted parties $(\mathcal{D}^*, \mathcal{E}^*)$. We consider four security guarantees, namely correctness, secrecy, robustness, and fairness. This is modeled with four adversary structures $\mathcal{Z}^c$, $\mathcal{Z}^s$, $\mathcal{Z}^r$, and $\mathcal{Z}^f$, one for each security requirement[4]: Correctness is guaranteed for $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{Z}^c$, secrecy is guaranteed for $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{Z}^s$, robustness is guaranteed for $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{Z}^r$, and fairness is guaranteed for $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{Z}^f$. We have the assumption that $\mathcal{Z}^r \subseteq \mathcal{Z}^c$ and $\mathcal{Z}^f \subseteq \mathcal{Z}^s \subseteq \mathcal{Z}^c$, as secrecy and robustness are not well defined without correctness, and as fairness cannot be achieved without secrecy.

Our protocol for general adversaries is based on [HMZ08], which is an adaptation of the perfectly secure protocol of [Mau02] to the statistical case. For a generic protocol construction, it is sufficient to consider two parameters [HLMR11]: First, the state that is held in the protocol is defined in terms of a parameter that influences the secrecy. This parameter is the sharing parameter $\mathcal{S}$, a collection of subsets of $\mathcal{P}$ that defines which party obtains which values. Second, the reconstruct protocol is expressed in terms of an additional parameter determining the amount of error correction taking place. This parameter is the reconstruction parameter $\mathcal{R}$. In contrast to the perfect case, here we need to

[4] Since all our protocols achieve agreement on abort for free, we do not introduce a separate structure for this security property.

consider both active and passive corruption. Therefore, the reconstruction parameter is a monotone collection of pairs $(\mathcal{D}, \mathcal{E})$ of subsets of $\mathcal{P}$ where $\mathcal{D} \subseteq \mathcal{E}$: If all errors can be explained with an adversary $(\mathcal{D}, \mathcal{E}) \in \mathcal{R}$, the errors are corrected and the protocol continues; otherwise it aborts. This implies that the protocol aborts only if the actual adversary is not in $\mathcal{R}$. Such aborts are global, i.e., if some subprotocol aborts, the entire protocol execution halts.

3.1 A Parametrized Protocol for General Adversaries

In the following, we present the parametrized subprotocols for general adversaries and analyze them with respect to correctness, secrecy, and robustness. The main result (including fairness) is discussed in Sec. 3.2. As a first step, we introduce *group commitments* which are a generalization of IC signatures that allow even passively-corrupted parties to reliably verify signatures even without an honest majority. We then use these group commitments to construct a verifiable secret-sharing scheme, and describe how to perform computations on shared values.

Group Commitments. As a first step, we introduce the notion of *group commitments*, which is a pair of protocols GroupCommit and GroupReveal. GroupCommit allows a group $\mathcal{G}$ to commit to a value v on which they agree (while providing secrecy with respect to the remaining parties $\mathcal{P} \setminus \mathcal{G}$), and GroupReveal allows them to reveal this value to the remaining parties. Our definitions and protocols for group commitments are based on the IC signatures introduced in Sec. 2.

Definition 1 (IC Group Commitment). *A group $\mathcal{G}$ is* IC group committed *(or simply* committed*) to a value v, denoted by $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$, if for all pairs $(p_i, p_j) \in \mathcal{G} \times \mathcal{G}$, v is IC-signed with $\langle v \rangle_{i,j}$.*

Note that a default group commitment $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$ can be generated given that all parties in $\mathcal{P}$ know the value v. Furthermore, if all parties in $\mathcal{G}$ are actively corrupted, then any values held by correct parties constitute a valid group commitment. Additionally, group commitments inherit linearity from the underlying IC signature scheme.

Protocol GroupCommit: Given a set $\mathcal{G}$ of parties that agree on a value v, compute a valid group commitment $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$ on v.

1. For each pair $(p_i, p_j) \in \mathcal{G} \times \mathcal{G}$ invoke IC-Sign on v with signer p_i and intermediary p_j.
2. If any invocation of IC-Sign outputs $\perp$, all parties output $\perp$. Otherwise, each party outputs the concatenation of the outputs of the invocations of IC-Sign.

Fig. 1. The group commit protocol for a group $\mathcal{G}$

Lemma 1. *Given a set $\mathcal{G}$ of parties that agree on a value v. If all parties in $\mathcal{G}$ are correct (i.e. $\mathcal{G} \cap \mathcal{D}^* = \emptyset$),* GroupCommit *correctly computes a valid group commitment $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$ on v. Otherwise,* GroupCommit *either correctly computes a valid group commitment $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$ on v, or all parties in $\mathcal{P}$ output $\perp$.* GroupCommit *is always secret and robust.*

Proof. Secrecy and robustness follow immediately by inspection. For correctness, we first have to show that if the protocol outputs a group commitment, then all signatures held by correct parties p_j are for the value v. This follows from the fact that IC-Sign always results either in a correct signature $\langle v \rangle_{i,j}$ or in $\perp$, even when the signer (or intermediary) is actively corrupted. Second, if all parties in $\mathcal{G}$ are correct, then it follows from the properties of IC-Sign that it never outputs $\perp$. □

If a group $\mathcal{G}$ is committed to a value v (e.g. if the GroupCommit protocol resulted in a valid group commitment and did not output $\perp$), the GroupReveal protocol reveals the value v to all parties in $\mathcal{P}$. During the protocol run, the adversary might be able to provoke conflicts that depend on the sets $\mathcal{D}^*$ and $\mathcal{E}^*$ of corrupted parties. Therefore, we introduce a parameter $\mathcal{R}$, which is a monotone collection of pairs $(\mathcal{D}, \mathcal{E})$ of subsets of the player set, where $\mathcal{D} \subseteq \mathcal{E}$: Whenever all conflicts in a given situation can be explained with an adversary $(\mathcal{D}, \mathcal{E}) \in \mathcal{R}$, the corresponding values are ignored (corrected), and the protocol proceeds; otherwise it aborts. Note that GroupReveal is the only subprotocol that might abort. All other protocols abort only if they use GroupReveal as a subprotocol. Therefore, it is sufficient to discuss agreement on abort only for this protocol.

We emphasize that the conflicts in GroupReveal do not only depend on the set $\mathcal{D}^*$ of actively corrupted parties, but also on the set $\mathcal{E}^*$ of passively corrupted parties, due to their inability to reliably verify IC-signatures. That means, in this protocol, even passive corruptions have a strong impact on correctness (and robustness).

Lemma 2. *Given the reconstruction parameter $\mathcal{R}$, the commitment group $\mathcal{G}$, and a group commitment $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$ for a value v,* GroupReveal *reveals v to all parties. The protocol is statistically correct if $\mathcal{G} \not\subseteq \mathcal{D}^*$ and*
$\forall (\mathcal{D}, \mathcal{E}) \in \mathcal{R}$:

$$\mathcal{G} \setminus \mathcal{D} \not\subseteq \mathcal{D}^* \ \vee \ (\mathcal{G} \not\subseteq \mathcal{E} \ \wedge \ \mathcal{P} \setminus \mathcal{E} \not\subseteq \mathcal{D}^*) \ \vee \ (\mathcal{G} \not\subseteq \mathcal{E}^* \ \wedge \ \mathcal{P} \setminus \mathcal{E}^* \not\subseteq \mathcal{D}).$$

The protocol is statistically robust if additionally $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{R}$, and always guarantees agreement on abort.

Proof. Correctness: Consider an actual protocol execution with correct value v and an adversary corrupting $(\mathcal{D}^*, \mathcal{E}^*)$. Denote with $\{\mathcal{V}_u\}$ the resulting collection of subsets of $\mathcal{P}$ in Step 3.

We first show that given the precondition $\mathcal{G} \not\subseteq \mathcal{D}^*$, we have

$$\big(\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_v) \subseteq \mathcal{D}^*\big) \ \wedge \ \big(\mathcal{G} \subseteq \mathcal{E}^* \ \vee \ \mathcal{P} \setminus \mathcal{V}_v \subseteq \mathcal{E}^*\big).$$

The precondition $\mathcal{G} \not\subseteq \mathcal{D}^*$ implies that there is at least one correct party $p_i \in \mathcal{G}$. In Step 1, this p_i broadcasts its value $u_i (= v)$ and invokes IC-Reveal on the

Protocol GroupReveal: Given the set $\mathcal{G}$ and a group commitment $\langle\!\langle v \rangle\!\rangle_{\mathcal{G}}$, reveal v to all parties.

1. For each party $p_i \in \mathcal{G}$:
   (a) p_i broadcasts v. Denote the broadcasted value with u_i.
   (b) For each party $p_j \in \mathcal{G}$: Invoke IC-Reveal on $\langle v \rangle_{j,i}$.
   (c) A party $p_k \in \mathcal{P} \setminus \mathcal{G}$ accepts u_i if all invocations of IC-Reveal output ("accept", u_i).
2. For each party $p_k \in \mathcal{P} \setminus \mathcal{G}$:
   (a) If p_k accepted at least one value in Step 1(c), and all accepted values are the same, then set u_k to this value. Else set $u_k := \perp$.
   (b) p_k broadcasts u_k.
3. Let $\mathcal{V}_u$ denote the set of parties that broadcasted u in Step 1(a) of 2(b), respectively. If $\exists (\mathcal{D}, \mathcal{E}) \in \mathcal{R}$ and a value v', such that
$$\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_{v'}) \subseteq \mathcal{D} \ \wedge \big(\mathcal{G} \subseteq \mathcal{E} \ \vee \ \mathcal{P} \setminus \mathcal{V}_{v'} \subseteq \mathcal{E}\big)$$
then output v'. Else abort.

Fig. 2. The group reveal protocol for a group $\mathcal{G}$

signatures $\langle v \rangle_{j,i}$ for $p_j \in \mathcal{G}$. It follows from the properties of IC-Reveal that all correct parties accept all these signatures. Hence, all correct parties in $\mathcal{P} \setminus \mathcal{G}$ accept the value $u_i (= v)$, and broadcast either v or $\perp$ in Step 2, but not a wrong value, i.e. $\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_v) \subseteq \mathcal{D}^*$. Furthermore, either $\mathcal{G} \subseteq \mathcal{E}^*$, or there is an honest party $p_j \in \mathcal{G}$. In the latter case, an actively corrupted $p_i \in \mathcal{G}$ can only forge the signatures $\langle v \rangle_{j,i}$ towards passively corrupted parties. Hence, it is guaranteed that all honest parties p_k broadcast the correct value $u_k = v$ in Step 2, and we have $\mathcal{P} \setminus \mathcal{V}_v \subseteq \mathcal{E}^*$.

Second, we show that given the precondition in the lemma, the protocol execution under consideration does not output an (incorrect) value $v' \neq v$, i.e., for all $v' \neq v$ and $(\mathcal{D}, \mathcal{E}) \in \mathcal{R}$ the condition in Step 3 is violated. To arrive at a contradiction, assume that for some $v' \neq v$ and $(\mathcal{D}, \mathcal{E}) \in \mathcal{R}$ it holds that $\big(\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_{v'}) \subseteq \mathcal{D}\big) \ \wedge \ \big(\mathcal{G} \subseteq \mathcal{E} \ \vee \ \mathcal{P} \setminus \mathcal{V}_{v'} \subseteq \mathcal{E}\big)$. From above, we have that $\big(\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_v) \subseteq \mathcal{D}^*\big) \wedge \big(\mathcal{G} \subseteq \mathcal{E}^* \vee \mathcal{P} \setminus \mathcal{V}_v \subseteq \mathcal{E}^*\big)$. Furthermore, by assumption we have that the precondition in the lemma is fulfilled. We split the proof according to which or-term of the second part of this precondition is fulfilled for the given $(\mathcal{D}, \mathcal{E})$:

Case $\mathcal{G} \setminus \mathcal{D} \not\subseteq \mathcal{D}^*$: Since $\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_{v'}) \subseteq \mathcal{D}$ and $\mathcal{G} \subseteq \mathcal{P}$, we have $\mathcal{G} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_{v'}) \subseteq \mathcal{D}$. It follows by inspection of the protocol that $\mathcal{G}$ and $\mathcal{V}_\perp$ are disjoint. Hence we have $\mathcal{G} \setminus \mathcal{V}_{v'} \subseteq \mathcal{D}$. Analogously, it follows from $\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_v) \subseteq \mathcal{D}^*$ that $\mathcal{G} \setminus \mathcal{V}_v \subseteq \mathcal{D}^*$. Therefore we have that $\mathcal{G} \subseteq \mathcal{D} \cup \mathcal{D}^*$, which is a contradiction to $\mathcal{G} \setminus \mathcal{D} \not\subseteq \mathcal{D}^*$.

Case $\mathcal{G} \not\subseteq \mathcal{E} \ \wedge \ \mathcal{P} \setminus \mathcal{E} \not\subseteq \mathcal{D}^*$: Since $\mathcal{G} \not\subseteq \mathcal{E}$, we have $\mathcal{P} \setminus \mathcal{V}_{v'} \subseteq \mathcal{E}$. Furthermore, we have that $\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_v) \subseteq \mathcal{D}^*$. It follows by inspection from the protocol that $\mathcal{V}_\perp$, $\mathcal{V}_{v'}$, and $\mathcal{V}_v$ are pairwise disjoint. Hence, we have that $\mathcal{P} \subseteq \mathcal{D}^* \cup \mathcal{E}$, which is a contradiction to $\mathcal{P} \setminus \mathcal{E} \not\subseteq \mathcal{D}^*$.

Case $\mathcal{G} \not\subseteq \mathcal{E}^* \ \wedge \ \mathcal{P} \setminus \mathcal{E}^* \not\subseteq \mathcal{D}$: This proof is identical to the previous case, with the only difference that $(\mathcal{D}^*, \mathcal{E}^*)$ is swapped with $(\mathcal{D}, \mathcal{E})$ and v with v'.

ROBUSTNESS: In the proof of correctness, we have shown that

$$\big(\mathcal{P} \setminus (\mathcal{V}_\perp \cup \mathcal{V}_v) \subseteq \mathcal{D}^*\big) \ \wedge \ \big(\mathcal{G} \subseteq \mathcal{E}^* \ \vee \ \mathcal{P} \setminus \mathcal{V}_v \subseteq \mathcal{E}^*\big).$$

Hence, given the correctness condition and $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{R}$, it follows immediately that the condition in Step 3 is fulfilled for the correct value v and $(\mathcal{D}^*, \mathcal{E}^*)$, i.e., that the protocol terminates without abort.
AGREEMENT ON ABORT: Since the abort decision is based only on broadcasted values, we always have agreement on abort. □

Given group commitments, protocols for sharing, reconstruction, addition, and multiplication can be constructed in a rather straightforward manner. Due to lack of space, the description of these protocols, as well as the proof of security of the parametrized protocol $\pi^{\mathcal{S},\mathcal{R}}$ (as stated in the following lemma) was moved to the full version of this paper.

Lemma 3. *Given the sharing specification $\mathcal{S}$ and the reconstruction parameter $\mathcal{R}$, the protocol $\pi^{\mathcal{S},\mathcal{R}}$ guarantees* statistical correctness *if*

$$\forall (\mathcal{D}, \mathcal{E}) \in \mathcal{R}, S, S' \in \mathcal{S} : \quad S \cap S' \neq \emptyset \quad \wedge \quad S \not\subseteq \mathcal{D}^* \quad \wedge$$
$$\big(S \setminus \mathcal{D} \not\subseteq \mathcal{D}^* \ \vee \ (S \not\subseteq \mathcal{E} \ \wedge \ \mathcal{P} \setminus \mathcal{E} \not\subseteq \mathcal{D}^*) \ \vee \ (S \not\subseteq \mathcal{E}^* \ \wedge \ \mathcal{P} \setminus \mathcal{E}^* \not\subseteq \mathcal{D})\big)$$

Furthermore, the protocol guarantees statistical secrecy *if additionally $\exists S \in \mathcal{S} : S \cap \mathcal{E}^* = \emptyset$, and/or* statistical robustness *if additionally $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{R}$.*

3.2 Main Result

The following theorem states the optimal bound for statistically secure MPC for general adversaries with both mixed adversaries and hybrid security. We show that the bound is sufficient for MPC by providing parameters for the generalized protocols described above. In the full version of this paper, we prove that the bound is also necessary.

Theorem 1. *In the secure channels model with broadcast and general adversaries, statistically secure (reactive) MPC among $n \geq 2$ parties with respect to $(\mathcal{Z}^c, \mathcal{Z}^s, \mathcal{Z}^r, \mathcal{Z}^f)$, where $\mathcal{Z}^r \subseteq \mathcal{Z}^c$ and $\mathcal{Z}^f \subseteq \mathcal{Z}^s \subseteq \mathcal{Z}^c$, is possible if $\mathcal{Z}^s = \{(\emptyset, \emptyset)\}$ or*

$$\forall (\cdot, \mathcal{E}^s), (\cdot, \mathcal{E}^{s\prime}) \in \mathcal{Z}^s, (\mathcal{D}^r, \mathcal{E}^r) \in \mathcal{Z}^r, (\mathcal{D}^c, \mathcal{E}^c) \in \mathcal{Z}^c \ :$$
$$\mathcal{E}^s \cup \mathcal{E}^{s\prime} \neq \mathcal{P} \quad \wedge \quad \mathcal{E}^s \cup \mathcal{D}^c \neq \mathcal{P} \quad \wedge$$
$$\Big(\mathcal{D}^c \cup \mathcal{D}^r \cup \mathcal{E}^s \neq \mathcal{P} \ \vee \ (\mathcal{E}^s \cup \mathcal{E}^r \neq \mathcal{P} \wedge \mathcal{D}^c \cup \mathcal{E}^r \neq \mathcal{P})$$
$$\vee \ (\mathcal{E}^s \cup \mathcal{E}^c \neq \mathcal{P} \wedge \mathcal{D}^r \cup \mathcal{E}^c \neq \mathcal{P})\Big)$$

This bound is tight: If violated, there are (reactive) functionalities that cannot be securely computed.

Proof (Sufficiency). If $\mathcal{Z}^s = \{(\emptyset, \emptyset)\}$, there is no secrecy requirement, and we can directly use the trivial non-secret protocol described in the Appendix of [HLMR11]. Otherwise, we employ the protocol $\pi^{\mathcal{S},\mathcal{R}}$ described in Sec. 3.1. We set $\mathcal{S} := \{\overline{\mathcal{E}^s} \mid (\cdot, \mathcal{E}^s) \in \mathcal{Z}^s\}$ and $\mathcal{R} = \mathcal{Z}^r \cup \mathcal{Z}^f$.

We apply Lemma 3 to derive correctness, secrecy and robustness: Given the bound in the theorem, the choice of the structures $\mathcal{S}$ and $\mathcal{R}$, and the fact that $(\mathcal{D}^*, \mathcal{E}^*)$ is an element of the corresponding adversary structure, it is easy to verify that the condition for each property is fulfilled. In particular, note that the correctness condition is also fulfilled for $(\mathcal{D}, \mathcal{E}) \in \mathcal{Z}^f$: Using that $\mathcal{Z}^f \subseteq \mathcal{Z}^s$, we have that $\mathcal{E}^s \cup \mathcal{E} \subseteq \mathcal{E}^s \cup \mathcal{E}^{s\prime} \neq \mathcal{P}$ (for some $\mathcal{E}^{s\prime}$) and $\mathcal{D}^c \cup \mathcal{E} \subseteq \mathcal{D}^c \cup \mathcal{E}^s \neq \mathcal{P}$ (where the inequalities follow from the second line of the condition in the theorem). This implies the condition for correctness.

Note that by our choice of $\mathcal{R}$, we have $\mathcal{Z}^f \subseteq \mathcal{R}$. Hence, for $(\mathcal{D}^*, \mathcal{E}^*) \in \mathcal{Z}^f$ the protocol is robust, and the adversary cannot abort. □

4 MPC with Threshold Adversaries

Trivially, the protocol for general adversaries can also be applied to the special case of threshold adversaries. Yet, protocols for general adversaries are super-polynomial in the number of parties for most adversary structures. Therefore, we present a protocol that exploits the symmetry of threshold adversaries, and is efficient in the number of parties.

The characterization for general adversaries (Sec. 3) can be adjusted for threshold adversaries: A mixed adversary is characterized by two thresholds (t_a, t_p), where he may corrupt up to t_p parties passively, and up to t_a of these parties even actively. The level of security (correctness, secrecy, robustness, and fairness) depends only on the number $(|\mathcal{D}^*|, |\mathcal{E}^*|)$ of actually corrupted parties. In the perfect setting [HLMR11], this is modeled with four pairs of thresholds, one for each security requirement, specifying the upper bound on the number of corruptions that the adversary may perform, such that the corresponding security requirement is still guaranteed. In the statistical setting, it follows from the bound for general adversaries that we need to consider multiple pairs of thresholds for each security guarantee. Consider the following example: Let $n = 6$ and $t_p^s = 2$. It is possible to obtain correctness for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (2, 6)$ and $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (3, 3)$, and robustness for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (1, 6)$ and $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (2, 3)$ in the same protocol. Yet, correctness and robustness cannot be guaranteed for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (3, 6)$ and $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (2, 6)$, respectively. Hence, this situation cannot be captured using only a single pair of thresholds for each security guarantee. Therefore, we introduce multi-thresholds T, i.e. collections of pairs of thresholds (t_a, t_p).

We consider the four multi-thresholds T^c, T^s, T^r, and T^f:[5] Correctness is guaranteed for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq T^c$,[6] secrecy is guaranteed for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq T^s$,

[5] As in the setting with general adversaries, we do not introduce a separate multi-threshold for agreement on abort.

[6] We write $(t_a, t_p) \leq T$ if $\exists (t_a', t_p') \in T : (t_a, t_p) \leq (t_a', t_p')$, where $(t_a, t_p) \leq (t_a', t_p')$ is a shorthand for $t_a \leq t_a'$ and $t_p \leq t_p'$.

robustness is guaranteed for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq T^r$, and fairness is guaranteed for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq T^f$. Again, we have the assumption that $T^r \leq T^c$ and $T^f \leq T^s \leq T^r$,[7] as secrecy and robustness are not well defined without correctness, and as fairness cannot be achieved without secrecy.

For threshold adversaries, we proceed along the lines of the general adversary case: We generalize the protocol of [FHM98, CDD+99] and introduce the *sharing parameter* d (corresponding to $\mathcal{S}$), and the *reconstruction parameter* E (corresponding to $\mathcal{R}$). Since we consider multi-thresholds, the reconstruction parameter E is a list of pairs (e_a, e_p) where $e_a \leq e_p$. Since for secrecy the actively corrupted parties $\mathcal{D}^*$ are not relevant, there cannot be two incomparable maximal adversaries. Hence, a single threshold is sufficient.

In this section, we assume that each party p_i is assigned a unique and publicly known evaluation point $\alpha_i \in \mathbb{F} \setminus \{0\}$. This implies that the field $\mathbb{F}$ must have more than n elements.

4.1 A Parametrized Protocol for Threshold Adversaries

In the following, we present the parametrized subprotocols and analyze them with respect to correctness, secrecy, and robustness. The main result (including fairness) is discussed in Sec. 4.2. The protocol is based on IC signatures as introduced in Sec. 2.

Verifiable Secret Sharing. The state of the protocol is maintained with a Shamir sharing [Sha79] of each intermediate result.

Definition 2 (d-Sharing). *A value s is d-*shared *when (1) there is a polynomial $\hat{s}(x)$ of degree d with $\hat{s}(0) = s$, and every party p_i holds a share $s_i = \hat{s}(\alpha_i)$, (2) for each share s_i, p_i holds a share polynomial $\hat{s}_i(y)$ of degree d with $\hat{s}_i(0) = s_i$, and every party p_j holds a share share $s_{ij} = \hat{s}_i(\alpha_j)$, and (3) for each share share s_{ij}, party p_i holds a signature $\langle s_{ij} \rangle_{j,i}$, and p_j holds a signature $\langle s_{ij} \rangle_{i,j}$. We denote a d-sharing of s with $[s]$, and the share s_i with $[s]_i$. A sharing parameter d is t-*permissive*, if the shares of all but t parties uniquely define the secret, i.e., $n - t > d$.*

Note that it follows from the linearity of Shamir sharings (i.e. a polynomial $\hat{s}(x)$ with $\hat{s}(0) = s$ where each party $p_j \in \mathcal{P}$ holds $\hat{s}(\alpha_j)$) and IC signatures, that d-sharings are linear.

Lemma 4. *Let $d < n$ be the sharing parameter. A d-sharing is secret if $|\mathcal{E}^*| \leq d$, and uniquely defines a value if d is $|\mathcal{D}^*|$-permissive.*

Proof. It follows directly from the properties of a polynomial of degree d that secrecy is guaranteed if the number $|\mathcal{E}^*|$ of (actively or passively) corrupted parties is at most d. Furthermore, $n - |\mathcal{D}^*| > d$ implies that there are at least $d + 1$ correct parties whose shares uniquely define a share polynomial. □

The share protocol takes as input a secret s from a dealer, and outputs a d-sharing $[s]$ (see Fig. 3).

[7] We write $T_1 \leq T_2$ if $\forall (t_a, t_p) \in T_1, \exists (t'_a, t'_p) \in T_2 : (t_a, t_p) \leq (t'_a, t'_p)$.

Protocol SHARE: Given input s from the dealer, compute a d-sharing $[s]$ of s.

1. The dealer chooses a random (bivariate) polynomial $g(x, y)$ with $g(0, 0) = s$, of degree d in both variables, and sends to each party $p_i \in \mathcal{P}$ the (univariate) polynomials $k_i(y) = g(\alpha_i, y)$ and $h_i(x) = g(x, \alpha_i)$.
2. For each pair of parties (p_i, p_j): p_i sends $k_i(\alpha_j)$ to party p_j, and p_j checks whether $k_i(\alpha_j) = h_j(\alpha_i)$. If this check fails, it broadcasts a complaint.
3. For all $k_i(\alpha_j)$, for which no inconsistency was reported, IC-SIGN is invoked once with signer p_j and intermediary p_i to compute the signature $\langle k_i(\alpha_j)\rangle_{j,i}$, and once with signer p_i and intermediary p_j to compute the signature $\langle k_i(\alpha_j)\rangle_{i,j}$.
4. The dealer broadcasts each value for which either an inconsistency was reported (Step 2), or the output of IC-SIGN was $\perp$ (Step 3), and a default signature is used.
5. If some party p_i observes an inconsistency between the polynomials received in Step 1 and the broadcasted values in Step 4, it accuses the dealer. The dealer answers the accusation by broadcasting both $k_i(y)$ and $h_i(x)$. Now, if some other party p_j observes an inconsistency between the polynomial received in Step 1 and these broadcasted polynomials, it also accuses the dealer. This step is repeated until no additional party accuses the dealer. For all broadcasted values, default signatures are used.
6. If the dealer does not answer some complaint or accusation, or if the broadcasted values contradict each other, the parties output a default d-sharing of a default value (with default signatures). Otherwise, each party p_i outputs the share $s_i := k_i(0)$, the share polynomial $\hat{s}_i(y) := k_i(y)$ with signatures $\langle \hat{s}_i(\alpha_j)\rangle_{j,i}$ (for $j = 1, \ldots, n$), and the share shares $s_{ji} := h_i(\alpha_j)$ with signatures $\langle s_{ji}\rangle_{j,i}$ (for $j = 1, \ldots, n$). The dealer outputs $\hat{s}(x) := g(x, 0)$.

Fig. 3. The share protocol for threshold adversaries

Lemma 5. *Let $d < n$ be the sharing parameter. On input s from the dealer,* SHARE *correctly, secretly, and robustly computes a d-sharing. If d is $|\mathcal{D}^*|$-permissive, and if the dealer is correct, the sharing uniquely defines the secret s.*

Proof. SECRECY: It follows from the properties of a bivariate polynomial that $g(x, y)$ reveals no more information about s than the specified output. After Step 1, the adversary does not obtain any additional information: In Step 4, a value s_{ij} is broadcasted only if p_i, p_j or the dealer is actively corrupted, i.e., the adversary knew the value already beforehand. Hence, the protocol does not leak more information than the specified output, and thus always provides secrecy.
CORRECTNESS: First, we have to show that the protocol outputs a valid d-sharing. Due to the bilateral consistency checks, any inconsistency in the values held by correct parties is detected in Step 2 and resolved in Step 4. Therefore, the values held by correct parties uniquely define a polynomial $g'(x, y)$ of degree d, which implies that $g'(x, 0)$ is of degree d. Furthermore, it follows from the properties of IC-SIGN that in Step 3, either a correct IC-signature is computed, or all parties output $\perp$. In the latter case, a default (and hence correct) IC-signature is used. Therefore, the output is a valid d-sharing. Second, we have to

show that if d is $|\mathcal{D}^*|$-permissive and if the dealer is correct, then the shared value equals the input of the dealer. A correct dealer can always consistently answer all complains and accusations with the correct values. Hence, if d is $|\mathcal{D}^*|$-permissive, the unique value defined by the sharing is the secret s.
Robustness: By inspection, the protocol does not abort. □

The public reconstruction protocol (Fig. 4) proceeds sharewise: For each share s_i, first party p_i broadcasts the share s_i together with the sharing polynomial $\hat{s}_i(y)$, and opens the signatures on all share shares $\hat{s}_i(\alpha_j)$. Second, all parties broadcast their share shares s_{ij}, and open the corresponding signatures. If active corruption took place, these two steps might produce conflicts between certain parties. Note that these conflicts do not only depend on the actively, but also on the passively corrupted parties, due to their inability to reliably verify IC-signatures. If these conflicts can be explained with an adversary corrupting $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq E$, then the share is accepted. Otherwise it is ignored. This technique allows also passively-corrupted parties to reliably verify signatures and therefore reconstruct the correct value. Finally, the secret is reconstructed using the accepted shares. Note that Public Reconstruction is the only subprotocol that might abort. All other protocols abort only if they use Public Reconstruction as a subprotocol and the invocation thereof aborts. Therefore, it is sufficient to discuss agreement on abort only for this protocol.

Lemma 6. *Given the sharing parameter d, the reconstruction parameter E, and a d-sharing $[s]$ of some value s,* Public Reconstruction *reconstructs s to all parties. The protocol is statistically correct if $|\mathcal{D}^*| < n - d$ and*

$$\forall (e_a, e_p) \in E: \ |\mathcal{D}^*| < n - d - e_a \ \vee \\ (d + e_p < n \ \wedge \ |\mathcal{D}^*| < n - e_p) \ \vee \ (|\mathcal{E}^*| < n - d \ \wedge \ |\mathcal{E}^*| < n - e_a).$$

Furthermore, it is statistically robust if additionally $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq E$, and always guarantees agreement on abort.

Proof. Correctness: The protocol outputs a value only if at least $d+1$ shares are accepted. Trivially, the output is correct if all accepted shares are correct, i.e., when incorrect shares are not accepted. More precisely, we have to show that for any incorrect share $s_i' \neq s_i$ and for each $(e_a, e_p) \in E$, the condition in Step 1(d) is violated. In this proof, we distinguish three cases, depending on which or-term of the condition in the lemma is fulfilled:

i. *Case $|\mathcal{D}^*| < n - d - e_a$:*
In order to broadcast a wrong share $s_i' \neq s_i$, an actively corrupted party p_i has to change the value of at least $n - d$ share shares. At least $n - d - |\mathcal{D}^*|$ of these share shares belong to correct parties that subsequently vote "no", i.e. $r \geq n - d - |\mathcal{D}^*|$. Since $|\mathcal{D}^*| < n - d - e_a$, this implies $r > e_a$, and the share is not accepted.

Protocol Public Reconstruction: Given a d-sharing $[s]$ of some value s, reconstruct s to all parties.

1. For each party p_i:
   (a) p_i broadcasts $\hat{s}_i(y)$ and invokes IC-Reveal on the signatures $\langle \hat{s}_i(\alpha_j) \rangle_{j,i}$ $(j = 1, \ldots, n)$ of all share shares.
   (b) Each p_j broadcasts its share share s_{ij} and invokes IC-Reveal on the corresponding signature $\langle s_{ij} \rangle_{i,j}$.
   (c) **Voting:** Each p_k checks whether
      i. the polynomial $\hat{s}_i(y)$ broadcasted in Step 1(a) is consistent with its share share, i.e. $s_{ik} = \hat{s}_i(\alpha_k)$,
      ii. the output of all invocations of IC-Reveal in Step 1(a) was "accept",
      iii. for all s_{ij} broadcasted in Step 1(b) either $s_{ij} = \hat{s}_i(\alpha_j)$ or the output of IC-Reveal on the corresponding signature $\langle s_{ij} \rangle_{i,j}$ was "reject".

   p_k broadcasts "yes" if all checks succeed, "no" if check i. or ii. fails, and $\perp$ otherwise. Let a and r denote the number of parties broadcasting "yes" and "no", respectively.
   (d) **Decision:** Accept s_i if $\exists (e_a, e_p) \in E : r \leq e_a \ \wedge \ (e_p + d \geq n \ \vee \ a \geq n - e_p)$. Otherwise ignore s_i.
2. **Output:** If at least $d + 1$ shares are accepted, interpolate these shares with a polynomial $\hat{s}'(x)$ and output $\hat{s}'(0)$. Otherwise abort.

Fig. 4. The public reconstruction protocol for threshold adversaries

ii. *Case* $d + e_p < n \ \wedge \ |\mathcal{D}^*| < n - e_p$*:*
Since $|\mathcal{D}^*| < n - d$, there are at least $d + 1$ correct parties. Hence, in order to broadcast a wrong share $s'_i \neq s_i$, an actively corrupted party p_i has to change the value of at least one share share belonging to a correct party. In Step 1(b), this correct party broadcasts the correct share share with a valid signature, and no correct party accepts the wrong share s'_i, i.e. $a \leq |\mathcal{D}^*|$. Since $|\mathcal{D}^*| < n - e_p$, we have $a < n - e_p$. Since we also have $d + e_p < n$, the share is not accepted.

iii. *Case* $|\mathcal{E}^*| < n - d \ \wedge \ |\mathcal{E}^*| < n - e_a$*:*
Since $|\mathcal{E}^*| < n - d$, there are at least $d + 1$ honest parties. Hence, in order to broadcast a wrong share $s'_i \neq s_i$, an actively corrupted party has to change the value of at least one share share belonging to an honest party, and to create the signature on this (incorrect) share share. All honest parties notice that this signature is not valid and reject, i.e., $r \geq n - |\mathcal{E}^*|$. Since $|\mathcal{E}^*| < n - e_a$, we have $r > e_a$, and the share is not accepted.

Robustness: Given that the correctness condition holds, the protocol guarantees robustness if enough (i.e. $d + 1$) shares are accepted. Let $(e_a, e_p) \in E$ such that $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (e_a, e_p)$. First, observe that if party p_i is correct, then $r \leq e_a$: All share shares and signatures broadcasted in Step 1(a) are correct and valid. Therefore, no correct party votes "no". Furthermore, if party p_i is honest, then $a \geq n - e_p$: If some p_j broadcasts a contradicting (wrong) share share in Step 1(b), then the signature on this share share is invalid for all honest parties.

It follows from the two observations above that shares from honest parties are always accepted. If $e_p + d < n$, then there are at least $d + 1$ honest parties and the protocol does not abort. Otherwise, if $e_p + d \geq n$, then also shares from correct parties are accepted. Since $|\mathcal{D}^*| < n - d$ there are always at least $d + 1$ correct parties and the protocol does not abort.

Agreement on abort: Since the abort decision is based only on broadcasted values, we always have agreement on abort. □

Addition, Multiplication, and Random Values. Linear functions (and in particular additions) can be computed locally, since d-sharings are linear: Given sharings $[a]$ and $[b]$, and a constant c, one can easily compute the sharings $[a]+[b]$, $c[a]$, and $[a] + c$. Computing a shared random value can be achieved by letting each party p_i share a random value r_i, and computing $[r] = [r_1] + \ldots + [r_n]$.

For the multiplication of two shared values, we first provide a non-robust multiplication protocol, which we then make robust using *dispute control* [BH06] and *circuit randomization* [Bea91]. Due to lack of space, the full description of the multiplication protocol was moved to the full version of this paper.

The Security of the Parametrized Protocol. Considering the security of the subprotocols described above, we can derive the security of the parametrized protocol, denoted by $\pi^{d,E}$:

Lemma 7. *Let d be the sharing parameter, and E be the reconstruction parameter, the protocol $\pi^{d,E}$ guarantees* statistical correctness *if $d < n - |\mathcal{D}^*|$, $2d < n$, and*

$$\forall (e_a, e_p) \in E: \quad |\mathcal{D}^*| < n - d - e_a \ \vee \\ (d + e_p < n \ \wedge \ |\mathcal{D}^*| < n - e_p) \ \vee \ (|\mathcal{E}^*| < n - d \ \wedge \ |\mathcal{E}^*| < n - e_a).$$

Furthermore, the protocol guarantees statistical secrecy *if additionally $|\mathcal{E}^*| \leq d$, and/or* statistical robustness *if additionally $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq E$.*

Proof. $\pi^{d,E}$ provides a certain security guarantee against $(|\mathcal{D}^*|, |\mathcal{E}^*|)$ if all subprotocols and the sharing provide this guarantee against $(|\mathcal{D}^*|, |\mathcal{E}^*|)$. For each guarantee, it can easily be verified that the condition in the lemma implies the conditions in the corresponding lemmas. □

4.2 Main Result

The following theorem states the optimal bound for statistically secure MPC for threshold adversaries with both mixed adversaries and hybrid security. We show that the bound is sufficient for MPC by providing parameters for the generalized protocols described above. The necessity of the bound follows directly from the corresponding proof for general adversaries that can be found in the full version of this paper.

Theorem 2. *In the secure channels model with broadcast and threshold adversaries, statistically secure (reactive) MPC among $n \geq 2$ parties with multi-thresholds T^c, T^s, T^r, and T^f, where $T^f \leq T^s \leq T^c$ and $T^r \leq T^c$, is possible if $T^s = \{(0,0)\}$ or*

$$\begin{aligned}
&\forall (t_a^c, t_p^c) \in T^c, (t_a^r, t_p^r) \in T^r, (\cdot, t_p^s), (\cdot, t_p^{s\prime}) \in T^s : \\
&\quad t_p^s + t_p^{s\prime} < n \quad \wedge \quad t_p^s + t_a^c < n \quad \wedge \\
&\quad \big(t_a^c + t_a^r + t_p^s < n \;\vee\; (t_p^s + t_p^r < n \wedge t_a^c + t_p^r < n) \\
&\qquad\qquad \vee\; (t_p^s + t_p^c < n \wedge t_a^r + t_p^c < n)\big)
\end{aligned}$$

This bound is tight: If violated, there are (reactive) functionalities that cannot be securely computed.

Proof (Sufficiency). If $T^s = \{(0,0)\}$, there is no secrecy requirement, and we can directly use the trivial non-secret protocol described in the Appendix of [HLMR11]. Otherwise, we employ the parametrized version $\pi^{d,E}$ of the protocol of [BGW88] described in Sec. 4.1 with $d := \widetilde{t_p^s}$ and $E := T^r \cup T^f$, where $\widetilde{t_p^s} = \max\{t_p^s \mid (\cdot, t_p^s) \in T^s\}$.

We apply Lemma 7 to derive correctness, secrecy and robustness: Given the bound in the theorem, the choice of the parameters d and E, and the fact that $(|\mathcal{D}^*|, |\mathcal{E}^*|)$ is below the corresponding threshold, it is easy to verify that the condition for each property is fulfilled. In particular, note that the correctness condition is also fulfilled for $(e_a, e_p) \in T^f$: Using that $T^f \leq T^s$, we have $d + e_p \leq 2\widetilde{t_p^s} < n$ and $e_a + e_p \leq t_a^c + d < n$ (where the inequalities follow from the second line of the condition in the theorem with $t_p^s = t_p^{s\prime} = \widetilde{t_p^s}$).

For fairness, note that $T^f \leq E$. Hence, for $(|\mathcal{D}^*|, |\mathcal{E}^*|) \leq (t_a^f, t_p^f)$ the protocol is robust, and the adversary cannot abort. □

5 Conclusion

Our results provide insights into the relations between passive corruption and different security requirements. The bounds presented in this work quantify the impact of passively corrupted parties on all security guarantees. We have shown that, in the statistical setting, passively corrupted parties play a significant role for all security guarantees, and not only for secrecy. Consider the following example: Let $n = 4$, $t_a^c = 2$, $t_p^c = 2$, $t_a^r = 1$, $t_p^r = 2$, and $t_p^s = 1$. For this choice of thresholds, the construction in this paper provides a protocol that is correct and robust (given that the adversary remains below the corresponding thresholds). Yet, we show that it is impossible to construct a protocol that tolerates a single additional passive corruption.

Furthermore, in addition to the known tradeoff between different security guarantees like robustness and correctness [HLMR11], we obtain a novel tradeoff between active and passive corruptions even when only considering a single security guarantee.

Solutions for the setting with general adversaries encompass all possible adversary structures. Yet, these protocols are usually superpolynomial in the number of parties. Therefore, protocols for the setting with threshold adversaries are of more practical relevance. In this work, we provide the first protocol allowing for multi-thresholds, a setting that is strictly more flexible than single-thresholds. This constitutes a substantial step towards general adversaries without losing efficiency.

References

[Bea89] Beaver, D.: Multiparty Protocols Tolerating Half Faulty Processors. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 560–572. Springer, Heidelberg (1990)

[Bea91] Beaver, D.: Efficient Multiparty Protocols Using Circuit Randomization. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992)

[BFH+08] Beerliová-Trubíniová, Z., Fitzi, M., Hirt, M., Maurer, U., Zikas, V.: MPC vs. SFE: Perfect Security in a Unified Corruption Model. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 231–250. Springer, Heidelberg (2008)

[BGW88] Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC 1988, pp. 1–10. ACM (1988)

[BH06] Beerliová-Trubíniová, Z., Hirt, M.: Efficient Multi-party Computation with Dispute Control. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 305–328. Springer, Heidelberg (2006)

[CCD88] Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols. In: STOC 1988, pp. 11–19. ACM (1988)

[CDD+99] Cramer, R., Damgård, I., Dziembowski, S., Hirt, M., Rabin, T.: Efficient Multiparty Computations Secure against an Adaptive Adversary. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 311–326. Springer, Heidelberg (1999)

[Cha89] Chaum, D.: The Spymasters Double-Agent Problem: Multiparty Computations Secure Unconditionally from Minorities and Cryptograhically from Majorities. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 591–602. Springer, Heidelberg (1990)

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

[FHHW03] Fitzi, M., Hirt, M., Holenstein, T., Wullschleger, J.: Two-Threshold Broadcast and Detectable Multi-party Computation. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 51–67. Springer, Heidelberg (2003)

[FHM98] Fitzi, M., Hirt, M., Maurer, U.: Trading Correctness for Privacy in Unconditional Multi-party Computation (Extended Abstract). In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 121–136. Springer, Heidelberg (1998)

[FHM99] Fitzi, M., Hirt, M., Maurer, U.M.: General Adversaries in Unconditional Multi-party Computation. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 232–246. Springer, Heidelberg (1999)

[FHW04] Fitzi, M., Holenstein, T., Wullschleger, J.: Multi-party Computation with Hybrid Security. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 419–438. Springer, Heidelberg (2004)

[GMW87] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC 1987, pp. 218–229. ACM (1987)

[HLMR11] Hirt, M., Lucas, C., Maurer, U., Raub, D.: Graceful Degradation in Multi-Party Computation (Extended Abstract). In: Fehr, S. (ed.) ICITS 2011. LNCS, vol. 6673, pp. 163–180. Springer, Heidelberg (2011)

[HM97] Hirt, M., Maurer, U.: Complete characterization of adversaries tolerable in secure multi-party computation. In: PODC 1997, pp. 25–34. ACM (1997)

[HMZ08] Hirt, M., Maurer, U.M., Zikas, V.: MPC vs. SFE: Unconditional and Computational Security. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 1–18. Springer, Heidelberg (2008)

[IKLP06] 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)

[Kat07] Katz, J.: On achieving the "best of both worlds" in secure multiparty computation. In: STOC 2007, pp. 11–20. ACM (2007)

[LRM10] Lucas, C., Raub, D., Maurer, U.: Hybrid-secure MPC: Trading information-theoretic robustness for computational privacy. In: PODC 2010, pp. 219–228. ACM (2010)

[Mau02] Maurer, U.M.: Secure Multi-party Computation Made Simple. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 14–28. Springer, Heidelberg (2003)

[PW92] Pfitzmann, B., Waidner, M.: Unconditional Byzantine Agreement for any Number of Faulty Processors. In: Finkel, A., Jantzen, M. (eds.) STACS 1992. LNCS, vol. 577, pp. 339–350. Springer, Heidelberg (1992)

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

[Sha79] Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)

[Yao82] Yao, A.C.: Protocols for secure computations (extended abstract). In: FOCS 1982, pp. 160–164. IEEE (1982)