Best Possible Information-Theoretic MPC

Shai Halevi[1]*, Yuval Ishai[2]**, Eyal Kushilevitz[2]**, and Tal Rabin[1]*

[1] IBM Research, USA
[2] Technion, Israel

Abstract. We reconsider the security guarantee that can be achieved by general protocols for secure multiparty computation in the most basic of settings: information-theoretic security against a semi-honest adversary. Since the 1980s, we have elegant solutions to this problem that offer full security, as long as the adversary controls a minority of the parties, but fail completely when that threshold is crossed. In this work, we revisit this problem, questioning the optimality of the standard notion of security. We put forward a new notion of information-theoretic security which is strictly stronger than the standard one, and which we argue to be "best possible." This notion still requires full security against dishonest minority in the usual sense, and adds a meaningful notion of information-theoretic security even against dishonest majority.

We present protocols for useful classes of functions that satisfy this new notion of security. Our protocols have the unique feature of combining the efficiency benefits of protocols for an honest majority and (most of) the security benefits of protocols for dishonest majority. We further extend some of the solutions to the malicious setting.

Keywords. Information-Theoretic Security, Secure Multiparty Computation

1 Introduction

In this work we revisit a question that seemed to be well understood since the 1980s: What is the best security guarantee that can be achieved by general protocols for secure multiparty computation in the simplest of all models? We put forward and study a new notion of information-theoretic security that provides a strictly stronger security guarantee than the standard notion. This security guarantee is in a sense the best possible. Before defining and motivating our new notion, we give some relevant background.

Protocols for secure multiparty computation (MPC) can be divided into two broad categories: *information-theoretic* MPC protocols, which offer unconditional

* Supported by the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236.

** Supported by ISF grant 1709/14, NSF-BSF grant 2015782, and a grant from the Ministry of Science and Technology, Israel and Dept. of Science and Technology, Government of India. Ishai was also supported by ERC grant 742754 (project NTSC).

security against computationally unbounded adversaries, and *computational* MPC protocols, which offer security against computationally bounded adversaries under standard cryptographic assumptions.

Information-theoretic MPC protocols not only provide unconditional security guarantees, but they are also typically simpler and have better concrete communication and computation costs than their computational counterparts. The efficiency gap can even grow with the number of parties by using efficient "packed secret-sharing" techniques [3, 18, 21], which divide the communication and computation costs between the parties (at the expense of slightly lowering the number of corruptions that can be tolerated).

A significant drawback of most information-theoretic MPC protocols, however, is that their security guarantees completely break down in the presence of a dishonest majority. Standard protocols, such as the so-called "BGW protocol" and its variants [6, 11, 35], allow a dishonest majority to learn the secret inputs of all parties. This is in contrast to computational protocols that can offer security even when all but one of the parties are dishonest. The above state of affairs gives rise to the following natural question:

> *Can we achieve the standard notion of information-theoretic security in the presence of an honest majority, while hiding the inputs of honest parties from a computationally unbounded dishonest majority?*

Classical negative results rule out information-theoretic protocols with the standard notion of security in the presence of a dishonest majority, even for a function as simple as the OR of n input bits [6, 13]. Unfortunately, these results further suggest that the answer to the question above may be negative. However, we observe that this *does not imply* that all inputs of the minority parties must be compromised, as is the case for existing protocols. This raises the possibility of finding a middle ground where only partial information about the inputs of honest parties is exposed.

Consider the following simple protocol for the OR function. Let G be a finite Abelian group, and let each party P_i locally map its input bit x_i to the group element $y_i = 0$ if $x_i = 0$ and to a uniformly random element $y_i \in G$ if $x_i = 1$. Now the parties run a secure addition protocol that computes the sum $Y = \sum_{i=1}^{n} y_i$ without revealing additional information to any subset of parties (even to a dishonest majority). Such addition protocol is easy to implement in the information-theoretic setting using the homomorphic property of additive secret sharing [7]. If $Y = 0$, then the parties output 0 and otherwise they output 1.

It is easy to see that the above protocol produces the correct output (i.e., the disjunction of the n inputs) except with $1/|G|$ error probability, which can be made arbitrarily small by choosing a large enough G. A key feature of this protocol is that even an adversary who corrupts a majority of the parties only learns limited information about the inputs of the uncorrupted parties, namely the OR of their input bits. This can provide in many cases a reasonable security guarantee. For instance, if the OR function is used to make a veto decision, then

the adversary can only learn whether *at least one* of the uncorrupted parties decided to veto, without learning additional information about the number or identity of parties who vetoed. This provides deniability even in the case where all but two of the parties are corrupted.

However, the above protocol fails to meet the standard security requirement for information-theoretic MPC in the presence of a dishonest minority, i.e. that a minority adversary learns nothing about the inputs of the uncorrupted parties as long as at least one of the adversary's inputs is $x_i = 1$. In this case, the (semi-honest) adversary can both learn the OR of the honest parties' inputs and force the output to be 1. Thus, there is room to do better.

1.1 Our Contribution

In this work we initiate a systematic study of the "best-possible information-theoretic security" for MPC protocols, when the adversary can corrupt an arbitrary number of parties. For the case of passive (semi-honest) adversary, we characterize the information that *must* be leaked to a dishonest majority. Then, restricting the adversary to learn *only* that amount of information would yield the best possible security that can be obtained in this setting. For some interesting functions, we also design information theoretic protocols that achieve this notion, namely provide standard security for honest majority, and leak only the necessary information to a dishonest majority.[3] We now give a more detailed account of our results.

New notion of security. We formally define our new notion of Best-possible Information-Theoretic MPC (BIT-MPC) as one that offers the standard notion of security against a corruption of a minority of parties and, additionally, offers the following kind of *residual security* against an adversary who corrupts a majority of the parties: the adversary cannot learn anything more than the *residual function* of the honest parties' inputs. By this, we mean that the adversary is allowed to learn only the value of the function on the inputs of the honest parties combined with *every choice of inputs* for the corrupted parties. In the case of OR from the above example (and similarly for the dual case of AND), this means that the adversary can only learn the OR of uncorrupted inputs, because the output for any choice of corrupted inputs can be derived from this information. As another example, consider the maximum function; in this case a dishonest majority can only learn the maximum of the honest parties' inputs.

Positive results. For some special functions of interest, we design protocols that realize the notion of BIT-MPC. This includes protocols for AND/OR, for deciding whether the inputs x_i satisfy a linear system of equations $Ax = b$ over

[3] Our notion of best-possible security, as well as both positive and negative results, apply not only in the threshold case but also to general adversary structures, replacing "honest majority" by any Q2 structure [29].

a finite field, and for computing functions like the maximum/minimum of the inputs, where the inputs come from a finite domain. While these functions are simple, they are useful for natural application scenarios. For instance, securely computing many parallel instances of AND can be useful for realizing multi-party instances of secure set intersection where sets come from a universe of bounded size. This in turn can be helpful for many real-world scenarios (consider a secure Doodle poll as a concrete example). Our protocols for these functions, especially when combined with other optimizations such as share-packing [18, 21] and pseudo-random secret sharing [14], lead to protocols that retain the efficiency advantages of honest-majority MPC and additionally offer a very meaningful protection against corrupted majorities. We expect such protocols to be attractive for implementations. See Section 6 for discussion of applications and concrete efficiency. Finally, most (but not all) of our results are easy to extend to the setting of security against *malicious* adversaries. This extension is discussed in Section 7.

Our BIT-MPC protocols build on protocols for non-interactive MPC (NIMPC) in the model of [4]. We rely on a restricted type of NIMPC protocols in which the correlated randomness is sampled uniformly from a linear vector space. We design such protocols for the above functions and show how to generally transform any such restricted NIMPC protocol into a BIT-MPC protocol.

Our results on NIMPC are independently motivated by the goal of making correlated randomness in NIMPC reusable or replacing it by a PKI setup under standard assumptions. It was previously known that both goals can be achieved for *general* functions by using indistinguishability obfuscation (see [26] and [25] respectively). Our work gives the first nontrivial examples for functions that admit NIMPC protocols with these useful features under weaker assumptions: one-way functions for reusability, and non-interactive key exchange (NIKE) for PKI setup.

Negative results. We complement our positive results by several negative results. First, by strengthening known results about characterizations of two-party secure computation (e.g., [5, 6, 12, 33]) and applying *partition arguments* (e.g., [12]), we show that our notion of BIT-MPC, indeed provides the best possible security. Namely, we prove that, for every (non-trivial) function f and for any coalition T, if standard security holds against the set of parties $\overline{T}$, then the parties in T must learn the corresponding residual function and, therefore, residual security is the best one can hope for.

Contrary to the general feasibility results for the standard notion of security, e.g. [6, 11], for our notion we rule out the possibility of *efficient* BIT-MPC protocols for all efficiently computable functions. More precisely, we show that such a positive result would imply that the polynomial hierarchy collapses. The proof of this fact is similar to the analogous negative result for best-possible indistinguishability obfuscation [24] (and implicitly in the context of instance hiding [1]). Our results do not rule out the possibility that *every* function f admits a BIT-MPC protocol if one does not take computational complexity into

account. This is the main question left open by our work. We do provide a first step towards resolving this question, by showing that such protocols exist for all 4-input functions (see Section 4.4).

Finally, we show a negative result that applies to a restricted class of protocols that captures most of our positives results. When considering Boolean functions f (outputting a single bit), protocols that have a certain "bilinear" structure over a finite field $\mathbb{F}$ are limited to only functions $f(x)$ that can be expressed as a linear test "$Ax = b$?" over $\mathbb{F}$. This relies on analogous results on the power of degree-2 randomized encodings of functions [31].

1.2 Related Work

Several prior works, including the works of Chaum [10], Ishai et al. [32], and Hirt et al. [28], provide a hybrid security guarantee of information-theoretic security for honest majority but need to switch to computational security against dishonest majority. This is contrasted with our work, where in both cases security is information-theoretic. Beyond the fact that we manage to preserve the information-theoretic setting, our results enjoy the efficiency benefits of this setting while the protocols in the hybrid model do not and, in fact, are even less efficient than their purely computational counterparts.

The problem of MPC with "residual security," extending the NIMPC model from [4] to the interactive setting, was recently considered in an independent work of Agrawal, Anand, and Prabhakaran [2]. Like our work, they show that residual security is the best possible in the presence of a corrupted majority. Furthermore, they give a combinatorial characterization for the class of functions for which residual security is *equivalent* to standard security. Similarly to our work, they also suggest a compiler from NIMPC to MPC, but their compiler is more restrictive than ours in that it does not allow interaction for emulating the NIMPC evaluator. Unlike our work, they do not consider the question of combining standard security for dishonest minority with residual security for dishonest majority, nor do they consider the class of functions to which our positive results apply.

2 Definitions

Notation. For a vector $v = (v_1, ...v_n)$ and $T = \{i_1, \ldots, i_{n'}\} \subseteq [n]$ a subset of size n' we define v_T to be $(v_{i_1}, ...v_{i_{n'}})$.

Our notion of Best-possible Information-Theoretically-secure MPC protocols (BIT-MPC) begins with the standard notion of secure protocols that only provides security against some sets of corrupted parties, but not other (e.g., only against a corrupted minority).[4] We augment the standard notion by requiring that even

[4] We note that in the pure information-theoretic setting we consider, the standard definitions below are equivalent to the definitions using a (computationally unbounded) simulator.

corrupted sets for which it is impossible to guarantee standard security, do not learn anything more than the *residual function* of the honest parties' inputs (which we later show is necessary). We start by recalling the definition of the residual function.

Definition 2.1 (Residual Function [27]). *Consider a fixed n-input function $f : (\{0,1\}^*)^n \to \{0,1\}^*$, let $x = (x_1, \ldots, x_n)$ be an input to f, and let $T = \{i_1, \ldots, i_{n'}\} \subseteq [n]$ be a subset of size n'. The* residual function *for T and x is an n'-input function $f_{T,x} : (\{0,1\}^*)^{n'} \to \{0,1\}^*$, obtained from f by restricting the input variables indexed by $[n] \setminus T$ to their values in x. That is, $f_{T,x}(y_1, \ldots, y_{n'}) = f(z_1, \ldots, z_n)$, where for $\ell \notin T$ we have $z_\ell = x_\ell$, while for $\ell = i_j \in T$ we have $z_\ell = y_j$.*

Definition 2.2 (Standard and Residual Security). *Let f be an n-input function, let $\Pi[\kappa]$ be an n-party protocol, for parties $P_1, \ldots, P_n$, that depends on a parameter κ, and fix some subset of parties $T \subseteq [n]$. Define $\mathsf{View}_{P_i}(x)$ as the local view of party P_i (including its randomness and the messages it received) during the execution of $\Pi(x)$.*

Standard security. *Π provides standard security against T if for any two inputs x, x' such that $x_T = x'_T$ and $f(x) = f(x')$, the two views $\mathsf{View}_T(x) = \{\{\mathsf{View}_{P_i}(x)\}_{i\in T},\ f(x)\}$ and $\mathsf{View}_T(x') = \{\{\mathsf{View}_{P_i}(x')\}_{i\in T},\ f(x')\}$ are statistically close, upto a distance of at most $2^{-\kappa}$.*

Residual security. *Π provides residual security against T if for any two inputs x, x' such that $x_T = x'_T$ and the residual function, $f_{T,x} \equiv f_{T,x'}$, the two views $\mathsf{View}_T(x) = \{\{\mathsf{View}_{P_i}(x)\}_{i\in T},\ f_{T,x}\}$ and $\mathsf{View}_T(x') = \{\{\mathsf{View}_{P_i}(x')\}_{i\in T},\ f_{T,x'}\}$ are statistically close, upto a distance of at most $2^{-\kappa}$.*

Definition 2.3 (BIT-MPC). *Let f be an n-input function, let $\Pi[\kappa]$ be an n-party protocol that depends on parameter κ, and consider some threshold $t \le n$. We say that Π is a t-private, best-possible, information-theoretic protocol for f (t-BIT-MPC) if the following conditions hold:*

- *Correctness: For all $x \in (\{0,1\}^*)^n$ it holds that $\Pi[\kappa](x) = f(x)$ with all but probability $2^{-\kappa}$ (taken over the randomness of Π).*
- *For any set $T \subseteq [n], |T| \le t$, Π provides standard security against T.*
- *For any set $T \subseteq [n], |T| > t$, Π provides residual security against T.*

We note that the definitions above were written in terms of an n-input/1-output function, but they extend naturally also to n-input/n-output functions (and later in this paper we sometimes need that extension). The only difference is that when considering a set $T \subseteq [n]$ we only look at the outputs of parties in the set T (i.e. $f(x)_T$). Hence, the residual function for T and x will be an n'-input/n'-output function, and the standard security notion will refer to every x, x' such that $f(x)_T = f(x')_T$ (even if $f(x) \neq f(x')$ when considering also the outputs outside T).

We also note that there is nothing special about threshold, and Definition 2.3 extends to any adversary structure (so, rather than considering t-BIT-MPC, we can talk about $\mathcal{T}$-BIT-MPC for an arbitrary adversary structure $\mathcal{T}$).

3 NIMPC with Restricted Correlated Randomness

The main technical tool that we use for our positive results on BIT-MPC are non-interactive MPC (NIMPC) protocols [4,20], where parties cannot interact with each other. To provide security in this setting, the parties are provided with some *correlated randomness*, which is chosen ahead of time, independently of the secret inputs. With this setup in hand, each party simply announces a single message to all parties and the output of the function is computed locally (possibly by all parties) on these messages.

Definition 3.1 (Non-Interactive MPC (NIMPC)). *A non-interactive MPC protocol $\Pi[\kappa]$ for n parties, $P_1, \ldots, P_n$, holding inputs $x = (x_1, \ldots, x_n)$ resp. (and parameter κ) is comprised of three parts:*

(1) randomness generation, $(r_1, \ldots, r_n) \leftarrow \mathsf{Gen}(\kappa)$, generating n random but correlated variables;
(2) local message functions, $\mathsf{Msg} = (\mathsf{Msg}_1, \ldots, \mathsf{Msg}_n)$, with Msg_i taking randomness r_i and local input x_i and outputting a message $m_i \leftarrow \mathsf{Msg}_i(x_i, r_i)$;
(3) evaluation function, $y \leftarrow \mathsf{Eval}(m_1, \ldots, m_n)$, taking n messages $\{m_i\}_i$ and computing the output y.

We define the view of a subset $T \subseteq [n]$ *in the execution of $\Pi[\kappa](x)$ as consisting of their own input and randomness, as well as everyone's messages,*

$$\mathsf{View}_T(x) = \{(x_i, r_i) | i \in T\} \cup \{m_1, \ldots, m_n\}.$$

We say that Π is a private non-interactive MPC protocol for an n-input function $f(x_1, \ldots, x_n)$ if the following conditions hold.

Correctness. *For any $x \in (\{0,1\}^*)^n$ it holds that $\Pi[\kappa](x) = f(x)$ with all but probability $2^{-\kappa}$ (taken over the randomness of Π).*

Privacy. *Π provides residual security against any subset. That is, for any set $T \subset [n]$ and any two inputs x, x' such that $x_T = x'_T$ and $f_{T,x} \equiv f_{T,x'}$, the two views $\mathsf{View}_T(x)$ and $\mathsf{View}_T(x')$ are statistically close, upto distance of at most $2^{-\kappa}$.*

When using an NIMPC protocol Π as a tool in our interactive setting, we must address the issues of how to generate the randomness, how messages are announced, and how to compute the output. It will be helpful to consider the following hierarchy of correlated randomness setups. Each level in the hierarchy has features that are useful independently of our main goal of constructing interactive BIT-MPC protocols.

1. **Unrestricted correlation:** Here the joint distribution of $(r_1, \ldots, r_n)$ output by Gen is arbitrary. This setting enables the strongest known results for NIMPC. In particular, every finite function f has a perfectly secure NIMPC protocol (as per Definition 3.1) in which the length of the messages is comparable to the truth-table size of f, and symmetric functions over $\{0,1\}^n$ have protocols in which the message length is quasi-polynomial in n [8].
2. **Linear correlation:** This is the type of setup most relevant to our work. A *linear correlation* is one that is uniform over some vector space $V \subset \mathbb{F}^m$ (for some $m \geq n$). More concretely, Gen is defined by a $k \times m$ matrix G over $\mathbb{F}$ and a partition of the m column indices of G into n sets S_i. The algorithm Gen proceeds by computing $r = sG$ for a random vector $s \in \mathbb{F}^k$ and letting r_i be r restricted to its S_i-entries. We will often let $m = n$ so that each r_i is a single field element.
3. **Replicated correlation:** This is a special case of linear correlation obtained by picking N random and independent field elements s_i and distributing each s_i to a fixed subset S_i of parties. An advantage of replicated correlations is that many copies of them can be generated by using a pseudo-random function. Thus, NIMPC with correlated randomness can be made *reusable* by using only a one-way function, or fast symmetric cryptography in practice. Using the share conversion technique from [14, 22], any n-party NIMPC protocol that uses a linear correlation setup can be compiled into one that uses the weaker replicated correlation setup, at the cost of increasing the size of the correlated randomness by at most a factor of 2^n.
4. **Pairwise-replicated correlation:** This is a special case of replicated correlation where each S_i is of size 2. Namely, each pair of parties share some secret randomness, independent of the randomness of other pairs. An advantage of this setup is that it can be implemented with a public-key infrastructure (PKI), using any 2-party non-interactive key agreement (NIKE) (which can be based on standard assumptions such as DDH). In contrast, replacing more general types of correlated or even replicated randomness by PKI is only known under stronger primitives such as multilinear maps or indistinguishability obfuscation [25].

Our BIT-MPC protocols will employ NIMPC protocols with linear correlations, which can be reduced to replicated correlations. Some useful special cases can be even based on NIMPC with pairwise correlations. Such special NIMPC protocols are independently motivated by the features discussed above.

4 Protocols

4.1 Compiler from NIMPC to BIT-MPC

Our main positive result is a compiler, that starts with an NIMPC protocol for a function f, and constructs a BIT-MPC protocol for f. In more detail, the ingredients for our compiler are protocols for parties $P_1, ..., P_n$:

- An n-party NIMPC protocol $\Pi = \{\mathsf{Gen}, \mathsf{Msg}, \mathsf{Eval}\}$ for an n-input function f;
- An n-party interactive MPC protocol Ψ_{Gen} for the randomized Gen function of Π.
- An n-party interactive MPC protocol Ψ_{Eval} for the Eval function of Π.

Given these protocols, the resulting interactive protocol $\Phi = \mathsf{Compile}(\Pi, \Psi_{\mathsf{Gen}}, \Psi_{\mathsf{Eval}})$ is as follows. On inputs $x_1, \ldots, x_n$ held by $P_1, \ldots, P_n$ resp.:

1. The parties run Ψ_{Gen} to evaluate Gen; r_i is the output of party P_i;
2. Each party P_i computes locally $m_i = \mathsf{Msg}_i(x_i, r_i)$;
3. The parties run Ψ_{Eval} each using m_i as its input to the protocol, to get $y = \mathsf{Eval}(m_1, \ldots, m_n)$.

Lemma 4.1. *Let f be an n-input function, Π a private NIMPC protocol for f, and let $\Psi_{\mathsf{Gen}}, \Psi_{\mathsf{Eval}}$, and the resulting $\Phi = \mathsf{Compile}(\Pi, \Psi_{\mathsf{Gen}}, \Psi_{\mathsf{Eval}})$ be as above.*

Correctness. *If Ψ_{Gen} and Ψ_{Eval}, are correct then Φ is a correct protocol for f.*

Security. *For any subset $T \subseteq [n]$, the following holds:*

Residual security. *If Ψ_{Gen} is correct and provides standard security against T, then Φ provides (at least) residual security against T.*

Standard security. *If Ψ_{Gen} is correct and Ψ_{Eval} is correct and provides standard security against T then the resulting Φ also provides standard security against T.*

Proof. Correctness can be verified by inspection. It remains to show security.

Residual security. The argument here is that, due to the security of Ψ_{Gen}, we are essentially in the world of NIMPC where members of T see only their own randomness and everyone's messages, hence we get (at least) residual security.

In more detail, since Ψ_{Gen} provides standard security against T, and as Gen has no secret inputs, then the transcript of Ψ_{Gen} does not reveal to the parties in T anything beyond their collective outputs, namely the (correlated) random values, $\{r_i : i \in T\}$.

Moreover, the transcript of Ψ_{Eval} is a randomized function of the inputs of that protocol, namely the m_i's, so at worst it reveals these m_i's to the parties in T. Hence, at worst, the view of the parties in T in the protocol $\Phi(x)$ consists of their own x_i, r_i's, and all the m_i's, which is exactly $\mathsf{View}_T(x)$ in Π, as defined in Definition 3.1. Similarly their view in $\Phi(x')$ is, at worst, $\mathsf{View}_T(x')$ in Π.

By the NIMPC security of Π, the views $\mathsf{View}_T(x)$ and $\mathsf{View}_T(x')$ are statistically close for any two x, x' with $x_T = x'_T$ and the same residual function relative to T, $f_{T,x} \equiv f_{T,x'}$.

Standard security. Here the argument is that (a) Gen is independent of the inputs, and (b) the transcript of Ψ_{Eval} does not leak to T anything about the inputs other than the function value.

Fix x, x' such that $x_T = x'_T$ and $f(x) = f(x')$, and denote the messages that the parties compute on these inputs by $m = (m_1, \ldots, m_n)$ and $m' = (m'_1, \ldots, m'_n)$,

respectively (i.e., $m_i = \mathsf{Msg}_i(x_i, r_i)$ and $m'_i = \mathsf{Msg}_i(x'_i, r_i)$). By correctness, we have that $\mathsf{Eval}(m) = \mathsf{Eval}(m')$ except with exponentially small probability. Since, by the locality of the messages in NIMPC protocols, we have $m_T = m'_T$, and as $\mathsf{Eval}(m) = \mathsf{Eval}(m')$, then the standard security against T of Ψ_{Eval} implies that the views of T in the executions $\Psi_{\mathsf{Eval}}(m)$ and $\Psi_{\mathsf{Eval}}(m')$ are statistically close.

Together with the fact that the protocol Ψ_{Gen} is independent of the inputs x, x', we conclude that the views of T in the executions $\Psi_{\mathsf{Eval}}(x)$ and $\Psi_{\mathsf{Eval}}(x')$ are also statistically close.

□

Using Lemma 4.1, we can deliver Best-possible Information-Theoretic MPC protocols in many interesting cases. Consider attempting a t-BIT-MPC for some function f, with a threshold $t < n/2$. By the lemma, all we need is some NIMPC protocol Π for f, together with:

- A protocol Ψ_{Eval} providing standard security against dishonest minority; and
- A protocol Ψ_{Gen} that provides complete privacy, even against dishonest majority, but only for the input-less function Gen.[5]

If we find an NIMPC protocol Π with a simple enough randomness generation function Gen, then we could hope to find a protocol Ψ_{Gen} with complete privacy. Adding a standard protocol for Eval (e.g., using the BGW construction), we would have standard security against dishonest minority, and residual security against dishonest majority, as needed. If we are willing to settle for a smaller threshold (say $t < 2n/5$), then we can use even more efficient protocols for Ψ_{Eval} (see, e.g., [18, 21]). This could give truly practical protocols, that provide meaningful (residual) security, no matter how many parties are corrupted.

Theorem 4.1. *Let $f(\cdot)$ be an n-input function. If there exists a private NIMPC protocol for f, $\Pi = \{\mathsf{Gen}, \mathsf{Msg}, \mathsf{Eval}\}$, and a protocol Ψ_{Gen} that computes Gen with standard security for all $T \subset [n]$ then, for any threshold $t < n/2$, there exists a t-private BIT-MPC protocol for f.* □

We remark again that there is nothing special about threshold, and an analogous theorem holds for any realizable adversary structure.

The main condition in Theorem 4.1 is that we have a protocol Ψ_{Gen} for the randomness-generation function with complete privacy. As discussed in Section 3, there is a hierarchy of correlated-randomness types that can enable the computation of different functions of increasing complexity. In the following, we examine various correlations that can be generated with complete privacy. The ideas behind some of the schemes that follow have been previously suggested but are presented here for self-containment and, even more so, because they are good examples for the application of our BIT-MPC theorem.

[5] We sometimes use the term *complete privacy* to refer to protocols that provide standard security against every subset $T \subseteq [n]$.

4.2 BIT-MPC From NIMPC with Pairwise Shared Randomness

One class of correlated randomness that can be generated with complete privacy (i.e., standard security against any set $T \subset [n]$), is pairwise shared randomness. The protocol Ψ_{Gen} is obvious: For $i < j$, party P_i sends to party P_j a random value $r_{i,j} \in \mathbb{Z}_p$, and the randomness of each P_i is set as $\{r_{k,i} : k < i\} \cup \{r_{i,k} : i < k\}$. It can easily be verified that this $\binom{n}{2}$-message protocol offers the desired security guarantees: Every party P_i sees only the values $r_{i,j}$ that it sent and $r_{j,i}$ that it received, and any value shared between two honest parties is not known to the attacker. Below, we examine some functions that can be computed using such pairwise shared randomness.

Shares of zero. Pairwise shared randomness can be easily converted into a correlated sharing of 0, just using local computation [9,15]: Each party P_i sets its share to $r_i = \sum_{k<i} r_{k,i} - \sum_{i<k} r_{i,k}$. It can be easily verified that, for any set of parties T, the only information revealed about the shares of the parties in $\overline{T}$ is their sum (and otherwise the shares of $\overline{T}$ are random).

Sum of inputs. Shares of zero are used in the following simple private NIMPC protocol for computing the sum [9,15] in a finite Abelian group.

Parties: $P_1, \ldots, P_n$
Input: $x_i \in G$ held by P_i
Output: $\sum_{i=1}^{n} x_i \in G$
Protocol:
Gen**:** Correlated sharing of 0; P_i has randomness r_i.
$\mathsf{Msg}_i(x_i, r_i)$**:** Output $m_i = x_i + r_i \in G$.
$\mathsf{Eval}(m_1, \ldots, m_n)$**:** Output $\sum_{i=1}^{n} m_i \in G$.

Fig. 1: Sum of Parties' Inputs in a finite Abelian group G

We remark that applying our BIT-MPC compiler to the NIMPC protocol in Fig. 1 is pointless, since the output of the SUM function by itself always exposes the residual function (i.e. the sum of the inputs of the honest parties), even to a dishonest minority. However, this NIMPC protocol will be a useful tool in compiling other functions into BIT-MPC.

Bitwise OR. Beimel et al. [4] present a private NIMPC for computing the OR function, assuming a (correlated randomness) sharing of 0. Each party chooses a new random value if its bit is 1 and uses the randomness from the zero-sharing if its input bit is 0. Then, the parties run the sum protocol on these values. If all the original bits are 0 then each party entered the randomness from the zero-sharing and thus the sum will be zero, and otherwise the sum will be nonzero with high probability. The parties output 0 if the sum is zero, and 1 otherwise. See Fig. 2.

Parties: $P_1, \ldots, P_n$
Input: $x_i \in \{0,1\}$ held by P_i
Output: $OR_{i=1}^{n} x_i$
Protocol:

Gen: Correlated sharing of 0, providing P_i with r_i.

$\mathsf{Msg}_i(x_i, r_i)$: Output: $m_i = \begin{cases} r_i & \text{if } x_i = 0 \\ R_i \in_R \mathbb{Z}_p & \text{otherwise} \end{cases}$

$\mathsf{Eval}(m_1, \ldots, m_n)$: *Output* $= \begin{cases} 0 & \text{if } \sum_{i=1}^{n} m_i = 0 \bmod p \\ 1 & \text{otherwise} \end{cases}$

Fig. 2: NIMPC protocol for the OR of Parties' Inputs

The above protocol exemplifies nicely how applying our BIT-MPC compiler and Theorem 4.1 adds privacy to the inputs of the parties. Observe that this protocol by itself reveals the OR of the honest parties' bits to the adversary, regardless of the number of corrupted parties and their inputs. (For example, a single corrupted P_j can check the equality $r_j \stackrel{?}{=} -\sum_{i \neq j} m_i$.) Applying our compiler, we improve security by ensuring that the sum of m_i's is never exposed to any minority group. In particular a single adversarial P_i with input value 1 learns nothing about the inputs of the other parties.

Computing the Maximum. To compute $MAX(x_1, \ldots, x_n)$ (with the x_i's taken from $[p]$ for some $p \in \mathbb{Z}$), each party i with input x_i locally computes the p bits $\chi_{i,\ell} := (x_i \geq \ell)$, for all $\ell \in [p]$. Then, the parties run p copies of the OR protocol, computing $\psi_\ell := OR_{i \in [n]} \chi_{i,\ell}$, for all ℓ. The maximum value is the largest index ℓ for which $\psi_\ell = 1$. See Fig. 3.

Lemma 4.2. *The protocol from Fig. 3 is a private NIMPC protocol for computing the maximum value of the inputs of the parties.* □

We remark that since we are dealing with semi-honest parties, then we do not have issues of consistency between the inputs in the different OR instances. The protocol from Fig. 3, though constant round, is inefficient for large p as it requires p invocations of the OR protocol and thus pn messages over all. This means that also the BIT-MPC protocol that we get by compiling it will be inefficient.

Although we do not know of a more efficient non-interactive MPC protocol from pairwise shared randomness, we are able to get a more efficient interactive BIT-MPC protocol for MAX. In the interactive setting, we can run the multiple copies of the OR protocol sequentially, rather than all at once, hence using binary search to get only $\log p$ invocations of the underlying OR protocol. See Fig. 4. Note that the bits ψ_j that are exposed by the protocol from Fig. 4 are actually implied by the output value $\max_i x_i$. We get:

Parties: $P_1, \ldots, P_n$
Input: $x_i \in \mathbb{Z}_p$ held by P_i
Output: $x^* = \max_i x_i$

Protocol:

Gen : generate p independent sharings of 0. The randomness of P_i is all its shares $r_i = \big(r_i[\ell]\big)_{\ell \in [p]}$.

$\mathsf{Msg}_i(x_i, r_i)$**:** 1. For each $\ell \in [p]$ compute the bit $\chi_{i,\ell} = \begin{cases} 1 & \text{if } x_i \geq \ell \\ 0 & \text{otherwise} \end{cases}$

2. Run p instances of the Msg function of the OR protocol, setting $\forall \ell \in [p],\ m_i[\ell] = \mathsf{Msg}^{OR}\big(\chi_{i,\ell},\ r_i[\ell]\big)$.

3. Concatenate $m_i = \big(m_i[\ell]\big)_{\ell \in [p]}$.

$\mathsf{Eval}(m_1, \ldots, m_n)$**:** for each $\ell \in [p]$ set $\psi_\ell = \mathsf{Eval}^{OR}\big(m_1[\ell], \ldots, m_n[\ell]\big)$. Output the largest ℓ such that $\psi_\ell = 1$.

Fig. 3: NIMPC protocol for MAX, Maximum of Parties' Inputs

Parties: $P_1, \ldots, P_n$
Input: $x_i \in \mathbb{Z}_p$ held by P_i
Output: $x^* = \max_i x_i$
Protocol:

The parties first run in parallel $\lceil \log p \rceil$ executions of a random sharing of zero. The randomness of P_i is all its shares $r_i = \big(r_i[\ell]\big)_{\ell \in [\log p]}$. Then, each party P_i does the following:

1. Set $min := 0, max := p - 1$.
2. For $j = 1, \ldots, \lceil \log p \rceil$ do the following:
3. Set $mid := \lceil (min + max)/2 \rceil$ and $\chi_{i,j} := (x_i \ \geq ?\ mid)$.
4. Run Msg_i of the OR protocol, broadcasting $m_{i,j} := \mathsf{Msg}_i^{OR}\big(\chi_{i,j}; r_i[j]\big)$.
5. Execute $\Psi_{\mathsf{Eval}}^{OR}(m_{i,j})$ to get $\psi_j := \mathsf{Eval}^{OR}\big(m_{1,j}, \ldots, m_{n,j}\big)$.
6. If $\psi_j = 1$ then $min := mid$, otherwise $max := mid - 1$.
7. End-for
8. Output min.

Fig. 4: A more efficient interactive BIT-MPC protocol for MAX

Lemma 4.3. *For any threshold $t \le n/2$, if $\Psi^{OR}_{\mathsf{Eval}}(m_{i,j})$ from Fig. 4 provides standard security against sets of size up to t, then the protocol from Fig. 4 is an interactive t-BIT-MPC for computing the MAX function.* □

4.3 BIT-MPC Based on Linearly-Correlated Randomness

As stated earlier, linear correlation is a powerful class of correlated randomness that can handle many interesting functions. The simplest format of linear correlation has each party P_i holds a piece of randomness r_i, where the vector $r = (r_1, \ldots, r_n)$ was chosen at random in some known linear subspace in $\mathbb{F}^n$. Namely $r = s \cdot A$, where A is a fixed, public $k \times n$ matrix that defines the linear space, and s is a uniformly random vector in $\mathbb{F}^k$.

This class generalizes the secret-sharing of zero that we used above, and we can compute the randomness generation for it with complete privacy, using similar techniques as for zero-sharing [7, 17]. Specifically, each party P_i chooses a random vector s_i and computes $t_i = s_i \times A$, then sends the entry $t_i[j]$ to P_j (for every j). The correlated-randomness element of each P_j is then set as $r_j = \sum_i t_i[j] = ((\sum_i s_i) \times A)[j]$.

To show complete privacy, fix a set $T \subset [n]$ of corrupted parties and note that the values $\{t_i[T] : i \notin T\}$ seen by the parties in T determine the s_i's only upto the solution of the system $(s_{\overline{T}} \times A)[T] = t_{\overline{T}}[T]$ (with $s_{\overline{T}}[T] = \sum_{i \notin T} s_i[T]$, $t_{\overline{T}}[T] = \sum_{i \notin T} t_i[T]$), which are exactly all the inputs with the same output at T.

We also note that using the share conversion technique from [14, 22], we can localy convert "replicated correlated randomness" to linearly correlated randomness. In a little more detail, by giving every subset $T \subset [n]$ a different random seed for a PRG/PRF, the parties can locally generate unbounded number of pseudo-random vectors in the range of $f_A(s) = s \cdot A$, without any interaction. This means that every party must keep 2^{n-1} seeds, but for small values of n this still yields a very practical way of generating linearly-correlated (pseudo)randomness, which can then be used in the protocols that we describe below.

Testing for membership in an affine space. We next show that linear correlations allow us to compute, for any matrix A, the function that determines whether an input vector belongs to the kernel of the rows of A. Namely

$$\mathsf{Affine}_{A,0}(x) = \begin{cases} 1 & \text{if } Ax = 0 \\ 0 & \text{otherwise} \end{cases}.$$

Since the parties have the vector $r = sA$ which is uniform in the columns space of A, it is sufficient to check if the inner product of x and r is zero. Hence each party computes $y_i = x_i r_i$, and the parties then run the SUM protocol from Figure 1.

This protocol can be modified to compute the function $\mathsf{Affine}_{A,b}(x)$, i.e., to check whether $Ax = b$ for a matrix A, as before, and a known vector b (rather than equality to zero). This is done by fixing a known vector w such that $Aw = b$

and having each party set $y_i = (x_i - w_i)r_i$ and run the SUM protocol. The resulting protocol is described in Figure 5.

Parties: $P_1, \ldots, P_n$
Parameter: Publicly known $A \in \mathbb{F}^{k \times n}$, $b \in \mathbb{F}^k$, and $w \in \mathbb{F}^n$ s.t. $Aw = b$.
Input: $x = (x_1, \ldots, x_n)$, $x_i \in \mathbb{F}$ held by P_i
Output: check whether $Ax = b$
Protocol:
Gen:
1. Linearly correlated randomness, $r = sA$ for a uniform $s \in \mathbb{F}^k$; P_i has r_i
2. Correlated sharing of 0, providing P_i with ρ_i. Let $R_i = (r_i, \rho_i)$.

$\mathsf{Msg}_i(x_i, R_i)$: Output $m_i = (x_i - w_i)r_i + \rho_i$

$\mathsf{Eval}(m_1, \ldots, m_n)$: Set $y = \Sigma_{i=1}^n m_i$, *Output* $= \begin{cases} 1 & \text{if } y = 0 \\ 0 & \text{otherwise} \end{cases}$

Fig. 5: NIMPC protocol for affine space membership, testing whether $Ax = b$ for public A, b,

Lemma 4.4. *The protocol from Fig. 5 is a private NIMPC protocol for affine space membership. Moreover, there exists a completely private protocol Π_{Gen} for computing the randomness generation function, with standard privacy against any set $T \subset [n]$.*

Proof. For correctness, note that $y = \langle r, x - w \rangle = sA(x - w) = \langle s, Ax - b \rangle$. Hence $y = 0$ with probability one when $Ax = b$, and $y \neq 0$ whp when $Ax \neq b$.

For privacy of the NIMPC protocol, we show that for any set $T \subset [n]$ and any two inputs x, x' such that $x_T = x'_T$ and $f_{T,x} \equiv f_{T,x'}$, the views of T on x and x' are distributed identically. It is convenient to first consider the case $b = 0$ (and thus w.l.o.g. $w = 0$). These views consist of (the public A and)

$r_T = \{r_i : i \in T\}$, $\rho_T = \{\rho_i : i \in T\}$, $x_T = \{x_i : i \in T\}$ or $x'_T = \{x'_i : i \in T\}$, *and* $m_{\overline{T}} = \{m_i = x_i r_i + \rho_i : i \notin T\}$ or $m' = \{m'_i = x'_i r_i + \rho_i : i \notin T\}$, *respectively.*

Since the ρ_i's are a random n-out-of-n sharing of zero, then the m_i's (or m'_i are uniformly random subject to their sum, regardless of r_T, ρ_T, x_T. It is thus enough to show that for any fixed r, we have $\sum_{i \notin T} m_i = \sum_{i \notin T} m'_i$ iff $f_{T,x} \equiv f_{T,x'}$.

To see this, notice that $f_{T,x} \equiv f_{T,x'}$ iff $A_{\overline{T}} x_{\overline{T}} = A_{\overline{T}} x'_{\overline{T}}$, where $A_{\overline{T}} x_{\overline{T}}$ is the sum of the columns of A corresponding to $i \notin T$, each multiplied by the corresponding x_i's (and similarly for $A_{\overline{T}} x'_{\overline{T}}$). Namely $A_{\overline{T}} x_{\overline{T}} =: \sum_{i \notin T} x_i A_i$ and $A_{\overline{T}} =: \sum_{i \notin T} x_i A_i$. This is true since, for any x^*_T, we have $f_{T,x}(x^*_T) = A_{\overline{T}} x_{\overline{T}} + A_T x^*_T$ and $f_{T,x'}(x^*_T) = A_{\overline{T}} x'_{\overline{T}} + A_T x^*_T$.

Consider therefore x, x' such that $A_{\overline{T}}x_{\overline{T}} = A_{\overline{T}}x'_{\overline{T}}$, and fix r and ρ. Then

$$\sum_{i\notin T} m_i - \sum_{i\notin T}\rho_i = \langle r_{\overline{T}}, x_{\overline{T}}\rangle = (sA)_{\overline{T}}x_{\overline{T}} = sA_{\overline{T}}x_{\overline{T}}$$
$$= sA_{\overline{T}}x'_{\overline{T}} = (sA)_{\overline{T}}x'_{\overline{T}} = \langle r_{\overline{T}}, x'_{\overline{T}}\rangle = \sum_{i\notin T} m'_i - \sum_{i\notin T}\rho_i,$$

and therefore $\sum_{i\notin T} m_i = \sum_{i\notin T} m'_i$. This completes the proof of privacy for the NIMPC protocol from Fig. 5 for the case of $b = 0$. The case of arbitrary b is similar (except that in the last equality we have another term $A_{\overline{T}}w_{\overline{T}}$ which is independent of x, x').

Finally, we note that the shared randomness in this protocol consists of $r = sA$ and a sharing of zero ρ, both of which can be computed with perfect privacy as we explained earlier. □

Corollary 4.1 (Affine Membership Over A Field). *For any fixed $A \in \mathbb{F}^{k\times n}$ and $b \in \mathbb{F}^k$, there is a $n/2$-BIT-MPC protocols for checking $Ax = b$ over $\mathbb{F}$.* □

Some applications of the affine membership protocol. Computing affine membership is more useful than it may seem. In particular, it captures most functions considered in the previous section as special cases, as well as additional useful functions. For example, the AND function can be realized utilizing the identity matrix $A = I$ and checking $Ax = b$ for $b = (1, \ldots, 1)$. The OR function is identical to AND up to relabeling of inputs and outputs, but can be realized directly using any invertible matrix A and $b = 0$, since $Ax = 0$ holds if and only if all the x_i's are 0. Affine membership can also be used to check equality of all inputs, namely the function $\mathsf{AllEq}(x_1, \ldots, x_n)$ which outputs 1 if $x_1 = x_2 = \ldots = x_n$ and 0 otherwise. Here we use a matrix $A \in \mathbb{F}^{(n-1)\times n}$ that reflects the equations $x_1 - x_2 = 0$, $x_2 - x_3 = 0$, …, $x_{n-1} - x_n = 0$, namely, the rows of A are all of the form $(0, \ldots, 0, 1, -1, 0, \ldots, 0)$. For this matrix A, the function $\mathsf{Affine}_{A,0}(x)$ is exactly $\mathsf{AllEq}(x)$.

4.4 Four-Input Functions

A somewhat surprising corollary of the per-subset nature of Lemma 4.1 is that any 4-input function can be computed with BIT-security against dishonest minority. Namely, we get standard security for a single corrupted party, and (at least) residual security for two or more corrupted parties.

Theorem 4.2. *For every 4-input function f, there is a 1-BIT-MPC interactive protocol for computing f.*

Proof. (Sketch) Let $\Pi = (\mathsf{Gen}, \mathsf{Msg}, \mathsf{Eval})$ be an NIMPC protocol for f with general correlated randomness (e.g., from [4]), and we describe interactive protocols for $\mathsf{Gen}, \mathsf{Eval}$ as needed.

- For Ψ_{Gen}, we use a 1-of-3 BGW protocol, run by P_2, P_3, P_4, to generate the needed correlated randomness.
- For Ψ_{Eval}, we use a 2-of-5 BGW protocol for evaluation, where P_2, P_3, P_4 each play a single party, and P_1 plays the role of two parties.

The reason that this construction works is that if there are three corruptions then the corrupted parties are allowed to learn the input of the honest party, so there is no security requirement. If there is only one corruption then the BGW protocols ensure standard security (even if the corrupted party is P_1 who plays a double role in the second BGW invocation).

It remains to show that we get (at least) residual security when two parties are corrupted. If the corrupted parties do not include P_1 then the 2-of-5 BGW protocol actually gives standard security. If P_1 is corrupted then two of P_2, P_3, P_4 are honest, hence we get standard security for the randomness generation step and therefore residual security for the combined protocol. □

5 Negative Results

5.1 When "Best-Possible" is the Best Possible

The first negative result justifies the term "best-possible" security, showing that in the information-theoretical regime, the security requirement against majority sets typically cannot be further strengthened. We start by considering two-party protocols and strengthen standard impossibility results for this setting (e.g., [5, 6, 12, 33]). Specifically, we show that in a two-party protocol between Alice and Bob to compute a function $f(x, y)$, standard information-theoretic security against Bob (i.e. if Bob learns the output and nothing else), implies that Alice necessarily learns Bob's input y. More concretely:

Lemma 5.1. *Let $f(x, y)$ be a boolean function and assume that each y is distinct; namely, for all $y \neq y'$ there exists an x such that $f(x, y) \neq f(x, y')$ (this is without loss of generality as otherwise Bob can pick one y from each "equivalence class"). Let $\mathcal{P}$ be any protocol where Bob learns $f(x, y)$ (with prob. 1) but no other information about x, then Alice can always identify y.*

Proof. Assume not. Then, for some pair of Bob inputs y, y' there is an Alice input x on which she cannot distinguish y from y'. Namely Alice's view (which consists of the transcript, as well as her input and randomness) on $(x, y), (x, y')$ is identically distributed and, in particular, it follows that $f(x, y) = f(x, y') = v$, for some $v \in \{0, 1\}$. Since y, y' are "distinct" then, for some other Alice input x', we have $f(x', y) \neq f(x', y')$. Since f is boolean then, without loss of generality, $f(x', y) = v$. Since Bob is assumed to learn nothing beyond the output, his distribution of views on (x, y) and on (x', y) (in particular, the distribution of transcripts) is the same. By a standard "corners lemma" (see, e.g., [12, 33]), it follows that the transcript on (x', y') is also distributed in the same way, contradicting the correctness of the protocol $\mathcal{P}$. □

Extensions. As stated, Lemma 5.1 assumes perfect correctness and perfect privacy. This however need not be the case and indeed some of the above papers (e.g., [12]) show that the same holds even when allowing ε-error and δ-privacy (i.e., where the statistical distance between the corresponding distributions is bounded by δ). The same modification applies in our case.

Another important extension is to deal with non-boolean functions f. Here, rather than asking for distinctness of the y's, we need a slightly more demanding (but still quite simple) richness requirement. Specifically, we ask that for any pair of Bob inputs y, y' that are not trivially distinguishable by Alice (i.e., where for some Alice input x we have $f(x,y) = f(x,y') = v$), there must also exists another Alice input x' for which $f(x',y) \neq f(x',y')$ and that one of these two values is equal to v. To illustrate this condition, consider the function $\mathsf{min}(x,y)$ (over some interval), where for all $y < y'$ the above condition holds with $x = y$ and $x' = y'$. This condition is a generalization of the distinctness property for boolean functions, used above; if this property holds then we will refer to the function f as being *non-trivial*. One can readily verify that the proof of Lemma 5.1 still holds for all non-trivial functions f (boolean or non-boolean).

Next, we deal with the case of n-input functions f, by applying a standard *partition argument*. It shows that, for any subset of parties T, if a protocol $\mathcal{P}$ satisfies standard security against a corrupted $\overline{T}$ then it can do no better than offering residual security against a corrupted T. (We usually think of T as a majority set, where it is always possible to ensure standard security against a corrupted $\overline{T}$, but the statement holds for any set T and this is important for generalizing the negative result to non-threshold access structures.) Formally,

Theorem 5.1. *Let f be an n-input function. Let $T \subset [n]$ be a subset of parties. Define the corresponding induced 2-argument function $f^T(\{x_i\}_{i\in T}, \{x_i\}_{i\notin T}) =: f(x_1,\ldots,x_n)$ and assume that f^T is non-trivial. Let $\mathcal{P}$ be any protocol where the parties in $\overline{T}$ learn $f(x_1,\ldots,x_n)$ but no other information about the input (information theoretically). Then, the parties in T learn the residual function $f_{T,x}$.*

Proof. Consider the two-party protocol $\mathcal{P}_T$ derived from $\mathcal{P}$ by Alice simulating the parties in T, Bob simulating the parties in $\overline{T}$ and together they compute the value $f^T(\{x_i\}_{i\in T}, \{x_i\}_{i\notin T})$. By the assumption on $\mathcal{P}$, Bob learns the output of f^T but nothing else. Hence, by Lemma 5.1 (and the following discussion), Alice learns Bob's input. In the terminology of the n-party protocol $\mathcal{P}$, this means that the view of the parties in T necessarily identifies the residual function $f_{T,x}$ (note that if two n-argument inputs x, x' induce the same residual function, i.e. $f_{T,x} = f_{T,x'}$, then they are mapped to equivalent inputs for the two-input function f^T). □

5.2 Efficient BIT-MPC Protocols are Rare

Our next goal is to show that *computationally efficient* BIT-MPC protocols are unlikely to exist even for simple families of functions. (Specifically, their existence would imply the collapse of the polynomial hierarchy.) We mimic similar results in the context of obfuscation [24] showing that if, for example, the family of 3-CNF formulas has efficient statistical-indistinguishability obfuscation, then the Polynomial Hierarchy collapses to its second level. This, in particular, relies on the following claim (implicit in [24]; see also [1] for a similar proof in the context of instance-hiding schemes):

Lemma 5.2. *Let $\mathcal{C}$ be a family of circuits where checking equivalence is co-NP complete (e.g., a simple family such as the family of all 3CNF formulae satisfies this). Assume that there exists a probabilistic polynomial time machine $\mathcal{S}$ that is given a circuit $C \in \mathcal{C}$ as its input and that its output is a probability distribution satisfying the following properties:*

- *If $C_1 \equiv C_2$ then $\mathcal{S}(C_1) \approx \mathcal{S}(C_2)$ (i.e., the statistical distance between $\mathcal{S}(C_1), \mathcal{S}(C_2)$ is bounded by some constant, say 1/3).*
- *If $C_1 \not\equiv C_2$ then $\mathcal{S}(C_1)$ and $\mathcal{S}(C_2)$ are far (i.e., the statistical distance between $\mathcal{S}(C_1), \mathcal{S}(C_2)$ is bounded from below by some constant, say 2/3).*

Then, the polynomial hierarchy collapses.

Consider the 3-input universal function $U_{\mathcal{C}}(C, x, \perp)$, where C is a boolean circuit from a family of circuits $\mathcal{C}$, as above, x is an input for the circuit C (and $\perp$ indicates that the third party has no input). We argue that there is no efficient BIT-MPC protocol for this function.

Theorem 5.2. *If there is a computationally efficient 3-party BIT-MPC protocol $\mathcal{P}$ for $U_{\mathcal{C}}$, then the polynomial hierarchy collapses.*

Proof. We use the protocol $\mathcal{P}$ to construct a machine $\mathcal{S}$, as required by Lemma 5.2. Doing so, the theorem follows.

We first turn $\mathcal{P}$ into a two-party protocol $\mathcal{P}'$ for computing the two-argument function $f(x, C) = U_{\mathcal{C}}(C, x, \perp)$, with Bob simulating the party in $\mathcal{P}$ holding the circuit C (i.e., a minority among the 3 parties) and Alice simulating the two others (i.e., the majority). By the best possible security of $\mathcal{P}$, Alice learns the function f_C and only this function (note that now the inputs of Bob are not distinct, as there may be several circuits that compute the same function f_C; an argument similar to that of Lemma 5.1 shows that Alice indeed learns f_C with any input x she may have).

We now construct $\mathcal{S}(C)$ as follows. Run $\mathcal{P}'$ on inputs (x_0, C), where x_0 is an arbitrary (fixed) input for Alice, and output her view in the protocol. On one hand, if $C_1 \equiv C_2$ then $f_{C_1} \equiv f_{C_2}$ and so Alice's view in both cases is identically distributed. On the other hand, if $C_1 \not\equiv C_2$, then Alice's view in the two cases is far apart. □

5.3 Simple BIT-MPC Protocols Have Limited Reach

Our last negative result shows that a natural class of "bilinear" NIMPC protocols, which captures most of our positive results, is limited in power. A *bilinear* NIMPC protocol over a finite field $\mathbb{F}$ is one in which the randomness $r_1, \ldots, r_n$ is linearly correlated, i.e., generated by applying a linear transformation A to a vector of random elements $(\rho_1, \ldots, \rho_m)$, and where each message m_i can be computed using a bilinear function $\mathsf{Msg}_i(x, r)$. That is, Msg_i is linear in both x (the inputs) and r (the randomness). Our goal is to show that "bilinear" protocols and, more generally, protocols where messages are computed as degree-2 polynomials in x and r, are limited in power. This negative result relies on a negative result for degree-2 randomizing polynomials from [31]. Concretely, [31] proved the following:

Lemma 5.3. *Suppose $f : \{0,1\}^n \to \{0,1\}$ admits a degree-2 randomized encoding over a finite field $\mathbb{F}$. Then, either of the following holds:*

- *f or its negation test a linear condition over $\mathbb{F}$; namely, are of the form $f_{A,b}(x) = 1$ iff $Ax = b$, for some $A \in \mathbb{F}^{\ell \times n}, b \in \mathbb{F}^{\ell}$; or*
- *f is a (deterministic) degree-2 polynomial.*

We observe that a bilinear protocol for a function f gives rise to a randomized degree-2 representation of f (of the first type). The degree is bounded by 2 because each message m_i is computed via a bilinear function $\mathsf{Msg}_i(x, r)$ and because r itself is a linear function of the underlying vector $(\rho_1, \ldots, \rho_m)$. The correctness and full robustness of the protocol imply that $m_1, \ldots, m_n$ encode $f(x)$ but give no other information about x. Thus, using Lemma 5.3, we get:

Theorem 5.3. *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function that has a bilinear NIMPC protocol over a finite field $\mathbb{F}$. Then, f or its negation test a linear condition over $\mathbb{F}$ (as defined above).*

6 Concrete Efficiency

In this section we make the case that our protocols, while restricted in the class of functionalities they apply to, can be useful for improving the concrete efficiency of natural secure computation tasks. We start by recalling some functions for which we present BIT-MPC protocols and discuss their relevance to natural secure computation tasks that can be motivated by real-world applications.

- AND/OR: Bitwise AND of long vectors of inputs can be used to realize multi-party Private Set Intersection (PSI) of sets over a universe of size $[N]$, where the input of each party is the length-N characteristic vector of its set. PSI has many real-world applications. One example is a secure Doodle poll, where the universe includes possible date and time slots, and each party's input is the subset of these slots in which he or she are available. See,

e.g., [34] and references therein for pointers on existing PSI protocols and their applications. Many of these applications are relevant even with feasible domain size, and even in the multiparty case.
- MIN/MAX: This is generally useful for auctions. Note that the variant in which the identity of the winner is revealed is reducible to the plain variant by encoding the identity of the owner of each input in the least significant bits of the input.
- Multiparty equality: Deciding whether all inputs are equal can be useful for checking whether there is agreement on the same candidate, whether different copies of the same information are identical, etc. In some of these cases, it is important to hide the identity of the outliers. See [19] for applications of secure two-party equality computation, some of which are relevant also in the multi-party case.

In all of the above cases, the residual security guarantee that we get in the presence of a dishonest majority is meaningful. In particular, it only reveals a very small amount of joint information about the inputs of honest parties, and moreover this information can typically be obtained in the ideal model via an adversarial choice of the input.

We now discuss the asymptotic and concrete efficiency features of optimized variants of our BIT-MPC protocols that make them more attractive than standard protocols for MPC with no honest majority. For concreteness we focus on the AND function, but similar optimizations apply to the other functions as well. We exclude from the discussion protocols based on fully homomorphic encryption (let alone general-purpose obfuscation) that do not seem to offer a competitive alternative for such simple computational tasks.

Existing "GMW-style" protocols for n-party AND that remain secure in the presence of an arbitrary number of (semi-honest) corrupted parties require $O(n^3)$ instances of oblivious transfer. This makes the total communication complexity $O(kn^3)$, where k is a computational security parameter. While some optimizations are possible using pseudo-random secret sharing (PRSS) [14], we are not aware of an OT-based protocol whose communication complexity is below $O(kn^2)$. In particular, even for a small number of parties such as $n = 10$, each party should communicate thousands of bits for a single AND computation. The main barrier is the use of oblivious transfers: the protocol consumes many of them, and efficient OT extension techniques [30] still require a significant amount of communication per OT instance.

Our BIT-MPC protocols replace OT with linear secret sharing, whose efficiency can be amortized via PRSS and/or share packing [21]. We note that the PRSS technique, when applied to threshold secret sharing schemes, incurs a computational cost (e.g., number of PRG invocations) that grows exponentially with the number of parties. Thus, this optimization can only be applied in practice when the number of parties is not too big.

Concretely, given a PRSS setup of replicated PRG seeds, our BIT-MPC protocol for AND of n input bits with 2^{-s} error probability needs only two

rounds of interaction, where each party broadcasts $s+1$ bits in each round (or sends a total of $(n-1)(s+1)$ bits over point-to-point channels). The computational complexity is dominated by roughly $\binom{n}{n/2}$ PRG calls (that can be implemented in practice via AES) per party. We do not know how to get MPC protocols that achieve a similar level of efficiency in the setting of standard MPC with no honest majority. Note that these efficiency advantages become very relevant when computing a large number N of instances. This case is motivated by some of the applications discussed above.

When the number of parties n is big, the PRSS technique no longer applies, but can be replaced by the use of packed secret sharing. This gives an amortized communication cost of $O(s)$ bits of point-to-point communication per AND computation per party, at the price of a slightly reduced (full) security threshold. Here one does not need any setup nor a direct implementation of broadcast to get this level of efficiency.

7 BIT-MPC with Security Against Malicious Parties

Our main focus in this paper is on BIT-MPC in the presence of a semi-honest (i.e., passive) adversary, who does not modify the messages sent by corrupted parties. In this section, we briefly discuss an extension of our notion of BIT-MPC and some of our results to the setting of a malicious (i.e., active) adversary.

We start by discussing the modified security definition for this case. In the case of security against a malicious adversary, we need to replace the direct definitions of standard and residual security, from Definition 2.2, by a simulation-based definition that compares the real-world execution of the protocol in the presence of a malicious adversary to an ideal-world execution in the presence of a simulator. Moreover, whereas in the case of an honest majority one can achieve full security (either when $t < n/3$ over secure point-to-point channels [6, 11] or with $t < n/2$ if broadcast is additionally available [35]), for the case of a dishonest majority we generally need to settle for "security with abort."

7.1 Defining BIT-MPC with a Malicious Adversary

At a high level, we modify the standard security definition of MPC (see [23]) by changing the ideal model experiment so that the adversary gets an explicit description of the residual function.[6]

For a set T of parties, we can consider four "types" of security that a protocol can offer against a corrupted T: ensuring either standard or residual security, and either guaranteed output delivery or security with abort. For simplicity,

[6] Clearly, this definition can only be satisfied with efficient simulation for functions whose residual function has a small description, such as functions on a small input domain or symmetric functions. Our negative results suggest that this restriction is inevitable.

the definition below only deals with two types of sets, "minority sets" against which we have full security with guaranteed output delivery, and "majority sets" against which we can only ensure residual security with abort. (Dealing with four different "types" is of course possible, but cumbersome.) Also for simplicity we only deal with the threashold variants of the definition (rather than arbitrary access structures). Hence, below we have a single threshold t upto which we ensure *full security* with guaranteed output delivery, whereas for more than t corrupted parties we settle for *residual security with abort*. The typical threshold t is $t < n/3$, for protocols over secure point-to-point channels, or $t < n/2$ with broadcast.

Definition 7.1. *Let f be an n-input function. let $\Pi[\kappa]$ be an n-party protocol that depends on parameter κ, and consider some threshold $t \leq n$. We say that Π is a t-secure, best-possible, information-theoretic protocol for f in the presence of malicious adversaries (or malicious t-BIT-MPC) if for every (malicious, static, computationally unbounded) adversary $\mathcal{A}$ attacking $\Pi[\kappa]$ there exists a simulator $\mathcal{S}$, with $2^{-\kappa}$ simulation error, that corrupts the same set of parties in the following ideal model:*

- **Standard security for up to t corruptions:** *If $\mathcal{A}$ corrupts at most t parties, the ideal model is as in the original definition from [23] for MPC with full security: each party sends its input to the trusted party (where the simulator $\mathcal{S}$ can change inputs of corrupted parties), the trusted party computes f and delivers the outputs to all parties.*
- **Residual security with abort beyond t corruptions:** *If $\mathcal{A}$ corrupts more than t parties, the ideal model is defined as follows: (1) each party sends its input to the trusted party; (2) the trusted party sends a description of the residual function of f (defined by the inputs of uncorrupted parties) to $\mathcal{S}$; (3) $\mathcal{S}$ decides whether to abort or to have the trusted party deliver the outputs of uncorrupted parties.*

7.2 BIT-MPC Protocols with Malicious Adversaries

In this section, we discuss the possibility of applying variants of the protocols from Section 4 in the presence of malicious adversaries. For this, we need to examine the effect of malicious behavior in all three components: the generation of the correlated randomness, the local computation of the NIMPC messages, and the distributed NIMPC evaluation. We discuss each of these components separately.

Correlated randomness generation. In the semi-honest case, we could generate any *linear correlation* n-securely using a simple information-theoretic protocol based on additive secret sharing. This protocol fails to be secure against malicious parties. In fact, the impossibility of information-theoretic coin-tossing with dishonest majority means that this insecurity is inherent. To get around this

impossibility, we consider the following "semi-malicious" relaxation of NIMPC with *replicated* correlation: when considering security against a collusion of the evaluator and a set of parties T, all of the random inputs involving T can be chosen adversarially (independently of the random inputs that are owned only by uncorrupted parties) but they are restricted to satisfy the prescribed replication pattern. It is easy to check that all of the previous protocols in this model remain secure even in this slightly more adversarial setting. Intuitively, this follows from the fact that even in the semi-honest model, the security of the uncorrupted parties is only protected by the random inputs that are unknown to the adversary. Finally, we observe that in the special case of *pairwise-replicated correlation*, we can generate the randomness in the straightforward way by making one of each pair of parties P_i, P_j pick the common randomness $r_{i,j}$ and send it to the other. Here the effect of a malicious adversary is equivalent to that of a semi-malicious adversary who can pick the random inputs adversarially but otherwise behaves honestly. Note that this is not the case for general replicated randomness, where the adversary can make replicated randomness owned by different honest parties inconsistent. From here on, we focus on BIT-MPC protocols that are obtained via NIMPC with pairwise-replicated randomness. This captures most of the examples from the previous section, including AND/OR and AllEq.

Local computation of NIMPC messages. Here we need to ensure that any malicious strategy of picking NIMPC messages by the adversary (independently of the honest parties' NIMPC messages) can be simulated by an honest strategy. Consider for example the direct protocol for the OR function. Here each party first maps a 0 input to 0 or a 1 input to a random nonzero group element, and then adds the correlated randomness (obtained via pairwise-replicated randomness). Note that for any fixed choice of the correlated randomness, every group element is a valid message, and moreover it is easy for the simulator to extract the input from the correlated randomness and the message (namely, the input is 0 if the two values are equal and 1 otherwise). One can check that the same is true for the more general NIMPC protocol for affine space membership. Here each party multiplies its (shifted) input by the correlated randomness r_i. Unless $r_i = 0$, which occurs with negligible probability, the simulator can extract an effective input from r_i and the NIMPC message.

Distributed NIMPC evaluation. This is the easiest part to handle, since we can simply apply off-the-shelf information-theoretic protocols that provide security against malicious adversaries. Depending on the setting, we can either use protocols such as [6, 11] for perfect t-security over secure point-to-point channels when $t < n/3$, or alternatively protocols such as [35] for statistical security over secure point-to-point channels and broadcast when $t < n/2$.

Beyond affine space membership. Using the above methodology, we can get BIT-MPC protocols for affine space membership whenever the correlated randomness can be obtained via pairwise-replicated correlation. This captures

the most useful examples of AND/OR and AllEq, but does not directly capture applications that build on top of them, such as the protocols for the MAX function. Recall that the MAX function computes the maximum of n integers in $\mathbb{Z}_p$. We presented two BIT-MPC protocols for MAX in Fig. 3 and Fig. 4. The first has constant round complexity but high communication complexity (linear in p), while the second uses binary search and multiple rounds to make the communication complexity grow only logarithmically with p. Both of these protocols use a BIT-MPC protocol for OR as a subroutine. However, they are both insecure against a malicious adversary even if the underlying OR protocol is fully secure against a malicious adversary. The attack is the same in both cases: even a single malicious party chooses its inputs for the OR protocol non-monotonically it can both simultaneously "win" (i.e., determine the output) and learn the maximum of the honest parties' inputs. This contradicts the full security requirement for the case of dishonest minority.

We propose two solutions to overcome the above attack and obtain a BIT-MPC protocol for MAX with security against malicious adversaries. The first solution is a sequential version of the protocol from Fig. 3, where in round ℓ the input for the OR function of party P_i is a bit $\chi_{i,\ell}$ which equals 1 if its input is at least $p - \ell$ and 0 otherwise. The protocol terminates with output $p - \ell_0$ after the first round ℓ_0 in which the OR-output is 1. In the protocol, the only degree of freedom the adversary has is to choose the first round in which one of its inputs is 1 (assuming that the protocol did not terminate before this round), and this choice can be simulated by an honest strategy. Finally, we note that it is also possible to get a constant-round protocol for MAX via a non-interactive reduction to secure modular addition that uses the nested subgroup technique from [16]. The idea is that MAX of inputs in $[m]$ can be reduced to addition in the group $\mathbb{Z}_{q^m}$ (where q is a prime of size $> 2^\kappa$) in the following way. Each input x_i is locally encoded as a random multiple of q^{m-x_i} in $\mathbb{Z}_{q^m}$, and then the n encoded inputs x'_i are added via a BIT-MPC protocol for addition in $\mathbb{Z}_{q^m}$. Due to the nested subgroup structure, the maximal multiple of q which divides the output will reveal the MAX value except with $1/q$ probability. Moreover, in this protocol a malicious adversary has no cheating space, as every possible choice of the encoded input x'_i in $\mathbb{Z}_{q^m}$ corresponds to an honest input. We leave open the question of obtaining a BIT-MPC protocol for MAX, with security against a malicious adversary, where the communication complexity grows logarithmically with m.

References

1. Abadi, M., Feigenbaum, J., Kilian, J.: On hiding information from an oracle (extended abstract). In: Aho, A. (ed.) 19th ACM STOC. pp. 195–203. ACM Press (May 1987) (Pages 4 and 19.)
2. Agarwal, N., Anand, S., Prabhakaran, M.: Brief announcement: On secure m-party computation, commuting permutation systems and unassisted non-interactive MPC. In: 45th International Colloquium on Automata, Languages, and Programming,

ICALP 2018, July 9-13, 2018, Prague, Czech Republic. pp. 103:1–103:4 (2018), https://doi.org/10.4230/LIPIcs.ICALP.2018.103 (Page 5.)
3. Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: Lightweight sublinear arguments without a trusted setup. In: CCS. pp. 2087–2104. ACM (2017) (Page 2.)
4. Beimel, A., Gabizon, A., Ishai, Y., Kushilevitz, E., Meldgaard, S., Paskin-Cherniavsky, A.: Non-interactive secure multiparty computation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 387–404. Springer, Heidelberg (Aug 2014) (Pages 4, 5, 7, 11, and 16.)
5. Beimel, A., Malkin, T., Micali, S.: The all-or-nothing nature of two-party secure computation. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 80–97. Springer, Heidelberg (Aug 1999) (Pages 4 and 17.)
6. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: 20th ACM STOC. pp. 1–10. ACM Press (May 1988) (Pages 2, 4, 17, 22, and 24.)
7. Benaloh, J.C.: Secret sharing homomorphisms: Keeping shares of a secret sharing. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 251–260. Springer, Heidelberg (Aug 1987) (Pages 2 and 14.)
8. Benhamouda, F., Krawczyk, H., Rabin, T.: Robust non-interactive multiparty computation against constant-size collusion. In: Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I. pp. 391–419 (2017), https://doi.org/10.1007/978-3-319-63688-7_13 (Page 8.)
9. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. Journal of Cryptology 1(1), 65–75 (1988) (Page 11.)
10. Chaum, D.: The spymasters double-agent problem: Multiparty computations secure unconditionally from minorities and cryptographically from majorities. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 591–602. Springer, Heidelberg (Aug 1990) (Page 5.)
11. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: 20th ACM STOC. pp. 11–19. ACM Press (May 1988) (Pages 2, 4, 22, and 24.)
12. Chor, B., Kushilevitz, E.: A zero-one law for Boolean privacy (extended abstract). In: 21st ACM STOC. pp. 62–72. ACM Press (May 1989) (Pages 4, 17, and 18.)
13. Chor, B., Kushilevitz, E.: A zero-one law for Boolean privacy. SIAM Journal on Discrete Math. 4, 36–47 (1991) (Page 2.)
14. Cramer, R., Damgård, I., Ishai, Y.: Share conversion, pseudorandom secret-sharing and applications to secure computation. In: Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings. pp. 342–362 (2005), https://doi.org/10.1007/978-3-540-30576-7_19 (Pages 4, 8, 14, and 21.)
15. Cramer, R., Damgård, I., Ishai, Y.: Share conversion, pseudorandom secret-sharing and applications to secure computation. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 342–362. Springer, Heidelberg (Feb 2005) (Page 11.)
16. Cramer, R., Fehr, S., Ishai, Y., Kushilevitz, E.: Efficient multi-party computation over rings. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 596–613. Springer, Heidelberg (May 2003) (Page 25.)
17. Damgård, I., Ishai, Y.: Constant-round multiparty computation using a black-box pseudorandom generator. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 378–394. Springer, Heidelberg (Aug 2005) (Page 14.)

18. Damgård, I., Ishai, Y.: Scalable secure multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 501–520. Springer, Heidelberg (Aug 2006) (Pages 2, 4, and 10.)
19. Fagin, R., Naor, M., Winkler, P.: Comparing information without leaking it. Commun. ACM 39(5), 77–85 (1996), http://doi.acm.org/10.1145/229459.229469 (Page 21.)
20. Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation (extended abstract). In: 26th ACM STOC. pp. 554–563. ACM Press (May 1994) (Page 7.)
21. Franklin, M.K., Yung, M.: Communication complexity of secure computation (extended abstract). In: 24th ACM STOC. pp. 699–710. ACM Press (May 1992) (Pages 2, 4, 10, and 21.)
22. Gilboa, N., Ishai, Y.: Compressing cryptographic resources. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. pp. 591–608 (1999), https://doi.org/10.1007/3-540-48405-1_37 (Pages 8 and 14.)
23. Goldreich, O.: Foundations of Cryptography: Basic Applications. Cambridge University Press (2004) (Pages 22 and 23.)
24. Goldwasser, S., Rothblum, G.N.: On best-possible obfuscation. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 194–213. Springer, Heidelberg (Feb 2007) (Pages 4 and 19.)
25. Halevi, S., Ishai, Y., Jain, A., Komargodski, I., Sahai, A., Yogev, E.: Non-interactive multiparty computation without correlated randomness. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III. pp. 181–211 (2017), https://doi.org/10.1007/978-3-319-70700-6_7 (Pages 4 and 8.)
26. Halevi, S., Ishai, Y., Jain, A., Kushilevitz, E., Rabin, T.: Secure multiparty computation with general interaction patterns. In: Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016. pp. 157–168 (2016), http://doi.acm.org/10.1145/2840728.2840760 (Page 4.)
27. Halevi, S., Lindell, Y., Pinkas, B.: Secure computation on the web: Computing without simultaneous interaction. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 132–150. Springer, Heidelberg (Aug 2011) (Page 6.)
28. Hirt, M., Lucas, C., Maurer, U., Raub, D.: Graceful degradation in multi-party computation (extended abstract). In: Fehr, S. (ed.) ICITS 11. LNCS, vol. 6673, pp. 163–180. Springer, Heidelberg (May 2011) (Page 5.)
29. Hirt, M., Maurer, U.M.: Player simulation and general adversary structures in perfect multiparty computation. Journal of Cryptology 13(1), 31–60 (2000) (Page 3.)
30. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. pp. 145–161 (2003), https://doi.org/10.1007/978-3-540-45146-4_9 (Page 21.)
31. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: 41st FOCS. pp. 294–304. IEEE Computer Society Press (Nov 2000) (Pages 5 and 20.)
32. Ishai, Y., Kushilevitz, E., Lindell, Y., Petrank, E.: Black-box constructions for secure computation. In: Kleinberg, J.M. (ed.) 38th ACM STOC. pp. 99–108. ACM Press (May 2006) (Page 5.)
33. Kushilevitz, E.: Privacy and communication complexity. In: 30th FOCS. pp. 416–421. IEEE Computer Society Press (Oct / Nov 1989) (Pages 4 and 17.)

34. Pinkas, B., Schneider, T., Weinert, C., Wieder, U.: Efficient circuit-based PSI via cuckoo hashing. In: Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III. pp. 125–157 (2018), https://doi.org/10.1007/978-3-319-78372-7_5 (Page 21.)
35. 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) (Pages 2, 22, and 24.)