Communication Lower Bounds for Statistically Secure MPC, with or without Preprocessing

Ivan Damgård*, Kasper Green Larsen**, and Jesper Buus Nielsen***

Computer Science. Aarhus University

Abstract. We prove a lower bound on the communication complexity of unconditionally secure multiparty computation, both in the standard model with $n = 2t + 1$ parties of which t are corrupted, and in the preprocessing model with $n = t + 1$. In both cases, we show that for any $g \in \mathbb{N}$ there exists a Boolean circuit C with g gates, where any secure protocol implementing C must communicate $\Omega(ng)$ bits, even if only passive and statistical security is required. The results easily extends to constructing similar circuits over any fixed finite field. This shows that for all sizes of circuits, the $O(n)$ overhead of all known protocols when t is maximal is inherent. It also shows that security comes at a price: the circuit we consider could namely be computed among n parties with communication only $O(g)$ bits if no security was required. Our results extend to the case where the threshold t is suboptimal. For the honest majority case, this shows that the known optimizations via packed secret-sharing can only be obtained if one accepts that the threshold is $t = (1/2 - c)n$ for a constant c. For the honest majority case, we also show an upper bound that matches the lower bound up to a constant factor (existing upper bounds are a factor $\lg n$ off for Boolean circuits).

1 Introduction

In secure multiparty computation (MPC) a set of n parties compute an agreed function on inputs held privately by the parties. The goal is that the intended result is the only new information released and is correct, even if t of the parties are corrupted by an adversary.

In this paper we focus on unconditional security where even an unbounded adversary cannot learn anything he should not, and we ask what is the minimal amount of communication one needs to compute a function securely. In particular: how does this quantity compare to the size of the inputs and to the circuit size of the function? Since one can always compute the function without security by just sending the inputs to one party and let her compute the function, an interesting question is what overhead in communication (if any) is

* Supported by the ERC Advanced Grant MPCPRO.

** Supported by a Villum Young Investigator grant and an AUFF starting grant. Part of this work was done while KGL was a long term visitor at the Simons Institute for Theory of Computing.

*** Supported by the Independent Research Fund Denmark project BETHE.

required for a secure protocol? An even harder question is if the communication must be larger than the circuit size of the function. Note that the questions only seem interesting for unconditional security: for computational security we can use homomorphic encryption to compute any function securely with only a small overhead over the input size.

There is a lot of prior work on lower bounding communication in interactive protocols, see for instance [Kus92, FY92, CK93, FKN94, KM97, KR94, BSPV99, GR03] (and see [DPP14] for an overview of these results). The previous work most relevant to us is [DPP14]. They consider a special model with three parties where only two have input and only the third party gets output, and consider perfect secure protocols. This paper was the first to show an explicit example of a function where the communication for a (perfectly) secure protocol must be larger than the input.

Later, in [DNOR16], a lower bound was shown on the *number of messages* that must be sent to compute a certain class of functions with statistical security. When the corruption threshold t is $\Theta(n)$, their bound is $\Omega(n^2)$. This of course implies that $\Omega(n^2)$ bits must be sent. However, we are intersted in how the communication complexity relates to the input and circuit size of the function, so once the input size become larger than n^2 the bound from [DNOR16] is not interesting in our context.

In [DNPR16], lower bounds on communication were shown that grow with the circuit size. However, these bounds only hold for a particular class of protocols known as gate-by-gate protocols, and we are interested in lower bounds with no restrictions on the protocol.

In [IKM$^+$13] the case of statistically secure 2-party computation with preprocessing is considered, where the parties are given access to correlated randomness at the start of the protocol. They show that the input size is (essentially) both an upper and a lower bound for the communication needed to compute a non-trivial function in this model, if one allows exponentially large preprocessed data. If one insists on the more practical case of polynomial size preprocessing, virtually all known protocols have communication proportional to the circuit size of the function. However, in [Cou18] it was shown (also for the 2PC case) that even with polynomial size preprocessed data, one can have communication smaller than the circuit size of the function, for a special class of so-called layered circuits.

1.1 Our results

In this paper, we prove lower bounds for the model with n parties of which t are passively and statically corrupted. The network is synchronous, and we assume that the adversary can learn the length of any message sent (in accordance with the standard ideal functionality modeling secure channels which always leaks the message length). We consider statistically secure protocols in both the standard model with honest majority, $n = 2t + 1$ and the preprocessing model where $n = t + 1$ is possible.

To understand our results, note first that any function can be computed insecurely by sending the inputs to one party and let her compute the function.

This takes communication S where S is the input size. What we show in both models is now that for any S, there exists a function f with input size S such that any protocol that evaluates f securely must communicate $\Omega(nS)$ bits. As mentioned, [DPP14] showed that such an overhead over the input size is sometimes required, we are the first to show that it grows with the number of players. So we see that security sometimes comes at a price, compared to an insecure solution.

However, we can say even more: we are able to construct functions f as we just claimed such that they can be evaluated by circuits of size $O(S)$. This means we also get the following: In both models, for any $g \in \mathbb{N}$ there exists a Boolean circuit C with g gates, where any protocol that evaluates C securely must communicate $\Omega(ng)$ bits. For the honest majority case, the result easily extends to constructing similar circuits over any fixed finite field. This shows that for all sizes of circuits, the $\Omega(n)$ overhead of all known protocols for maximal t is inherent. It is the first time it has been shown that there are circuits of all sizes which must suffer this $\Omega(n)$ overhead ([DNOR16] implies this result for circuits of size n).

The reader should note that since our result only talks about functions with linear size circuits, this leaves open the question of overhead over the circuit size when the circuit is much bigger than the inputs[1].

Our results extend to the case where the threshold t is suboptimal. Namely, if $n = 2t + s$, or $n = t + s$ for the preprocessing model, then the lower bound is $O(gn/s)$ and this shows that the improvement in communication that we know we can get for honest majority using so-called packed secret-sharing, can only be obtained if one accepts that the threshold t is $t = (1/2 - c)n$ for a constant c. In more detail, [DIK10] shows that for large n end even larger circuits of "sufficiently nice" shape, one can get a perfectly secure protocol with communication $\tilde{O}(g)$ for circuits with g gates (where the $\tilde{O}$ hides logarithmic factors in g and n). This protocol uses packed secret sharing which allows us to share a vector of $\Theta(n)$ field elements where each share is only one field element. We can therefore do $\Theta(n)$ secure arithmetic operations in parallel "for the price of one". This construction gives communication $\tilde{O}(g)$ but a corruption threshold much smaller than $n/2$. However, using the so-called *committee approach* (originally by Bracha but introduced for MPC in [DIK+08]), one can build a new protocol for the same function and similar complexity but now with threshold $t = (1/2 - c)n$ for an arbitrarily small constant c. Our results now imply that there is no way to improve the committee approach (or any other approach) to yield $t = (1/2 - o(1))n$: the circuits we build in this paper are indeed "nice enough" to be handled by the protocol from [DIK10], so any hypothetical improvement as stated would yield a protocol contradicting our lower bound.

[1] This is a much harder question of a completely different nature: for instance, if you are given a circuit to evaluate securely, there might exist a much smaller circuit computing the same function, so proving something on the overhead over the circuit size in general seems out of the question unless we are "magically" given the smallest circuit for the function in question.

For honest majority, we also show an upper bound that matches the lower bound up to a constant factor for all values of $t < n/2$. This is motivated by the fact that the existing upper bound from [DN07] is a factor $\lg n$ off for Boolean circuits. We do this by exploiting recent results by Cascudo et al. [CCXY18] on so-called reverse multiplication friendly embeddings.

For dishonest majority with preprocessing, an upper bound for $t = n-1$ was already known. Namely, by an easy generalization of the two party protocol from [IKM$^+$13] (already mentioned there), one obtains communication complexity $O(nS)$ for any function where S is the input size, using an exponential amount of preprocessed data. This matches our lower bound up to a constant factor: for the functions we consider, circuit and input size are essentially the same, so our bound is $\Omega(ng) = \Omega(nS)$. This settles the question of communication complexity in the preprocessing model for maximal t and exponential size preprocessing. For the case of suboptimal values of t where $t = n - s$ we show an upper bound $O(tg/s)$ with polynomial size preprocessing, using a simple generalization of known protocols. We do not know if this can be strengthened to $\Omega(St/s)$ if one allows exponential size preprocessing.

On the technical side, what we show are actually lower bounds on the entropy of the messages sent on the network when the inputs have certain distributions. This then implies similar bounds in general on the average number of bits to send: an adversary who corrupts no one still learns the lengths of messages, and must not be able to distinguish between different distributions of inputs. Hence message lengths cannot change significantly when we change the inputs, otherwise the protocol is insecure.

To show our results, we start from a lower bound for the communication complexity of private information retrieval with or without preprocessing and one server. While such a bound follows from the results in [IKM$^+$13], we give our own (much simpler) proof for self-containment. From this bound we show lower bounds for honest majority in the 3-party case and then finally "lift" the results to the multiparty case, while for dishonest majority we go directly from 2-party to multiparty. The observations we make in the 3-party case are related, at least in spirit, to what was done in [DPP14], indeed we also prove a lower bound for a case where 2 parties have input and the third has output. There are two important differences, however: first, we prove results for statistical security which is stronger than perfect security as in [DPP14] (because we show lower bounds). Second, while [DPP14] considers a very general class of functions, we consider a particular function (the inner product) which makes proofs simpler, but more importantly, we need the structure of this function to lift our results to the multiparty case.

The lifting is done using a simple but effective trick which is new to the best of our knowledge: loosely speaking, we start from a circuit computing, say $f(x_1, .., x_n)$ where the x_i's are the private inputs. Then we introduce an extra input bit b_i for P_i, and demand that her output be $b_i \cdot f(x_1, ..., x_n)$. By a reduction to the 3-party case, we can show that P_i must communicate a lot when $b_i = 1$ and $b_j = 0$ for $j \neq i$. Since now the identity of the party who gets the output is

determined by the inputs, a secure protocol is not allowed to reveal this identity, and this forces all players to communicate a lot.

2 Preliminaries

2.1 Information Theory

We first recall the well-known Fano's inequality which implies that for a random variable X, if we are given the value of another random variable X' which is equal to X except with probability δ, then the uncertainly of X drops to 0 as $\delta \to 0$:

Lemma 1. *Let δ be the probability that $X \neq X'$ and $\mathcal{X}$ be the support set of X and X'. Then $\mathrm{H}(X \mid X') \leq h(\delta) + \delta(\lg|\mathcal{X}| - 1)$, where $h()$ is the binary entropy function.*

It is easy to see from this result that if δ is negligible in some security parameter while $\lg|\mathcal{X}|$ is polynomial, then $\mathrm{H}(X \mid X')$ is also negligible.

In the following we will use $\mathrm{D}(X, X')$ to denote the statistical distance between the distributions of X and X' with common support $\mathcal{X}$, that is:

$$\mathrm{D}(X, X') = \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr(X = x) - \Pr(X' = x)|$$

Now, from Lemmas 4.5 and 4.6 in [DPP98] it follows immediately that we can bound the change in entropy in terms of the distance;

Lemma 2. $|\mathrm{H}(X) - \mathrm{H}(X')| \leq \mathrm{D}(X, X')(\lg \mathcal{X} - \lg \mathrm{D}(X, X'))$

The other result we need considers a case where we have two random variables X, Y and another pair X', Y' such that $\mathrm{D}((X, Y), (X', Y'))$ is bounded by some (small) δ. Then we can show that $\mathrm{H}(X \mid Y)$ is close to $\mathrm{H}(X' \mid Y')$:

Corollary 1. *Assume $\mathrm{D}((X, Y), (X', Y')) \leq \delta$, and let $\mathcal{XY}$ be the support set of X, Y. Then we have $|\mathrm{H}(X \mid Y) - \mathrm{H}(X' \mid Y')| \leq 2\delta(\lg|\mathcal{XY}| - \lg \delta)$*

Proof. By the triangle inequality, it is easy to see that

$$\mathrm{D}(Y, Y') \leq \mathrm{D}((X, Y), (X', Y')) \ .$$

Now we can use the above lemma and the triangle inequality again to calculate as follows:

$$\begin{aligned} |\mathrm{H}(X|Y) - \mathrm{H}(X'|Y')| &= |\mathrm{H}(X, Y) - \mathrm{H}(Y) - (\mathrm{H}(X', Y') - \mathrm{H}(Y'))| \\ &\leq |\mathrm{H}(X, Y) - \mathrm{H}(X', Y')| + |\mathrm{H}(Y) - \mathrm{H}(Y')| \\ &\leq \delta(\lg|\mathcal{XY}| - \lg \delta) + \mathrm{D}(Y, Y')(\lg|\mathcal{Y}| - \lg \mathrm{D}(Y, Y')) \\ &\leq 2\delta(\lg|\mathcal{XY}| - \lg \delta) \ . \end{aligned}$$

□

Again we can see that if δ is negligible in a security parameter while $|\mathcal{XY}|$ is polynomial, then the difference in conditional entropies is negligible.

2.2 Unconditionally Secure MPC

We look at a special case of MPC called secure function evaluation. There are n parties $\mathsf{P}_1, \ldots, \mathsf{P}_n$. They are connected by secure point-to-point channels in a synchronous network. Each of them has an input $x_i \in \{0,1\}^I$ in round 1. Eventually each P_i gives an output $y_i \in \{0,1\}^O$. We assume that $t < n/2$ of the parties can be corrupted. We consider only passive security. In this setting security basically means that the outputs are correct and that the distribution of the view of any t parties and be sampled given only their inputs and outputs.

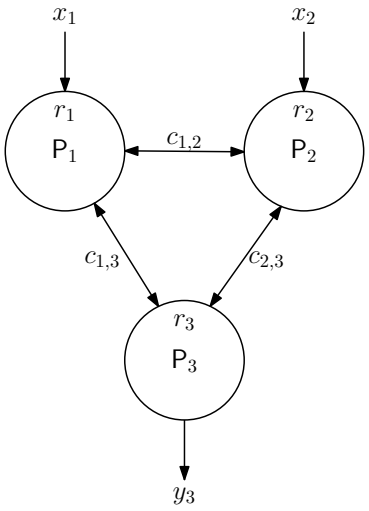

Fig. 1. A special case of the model where $n = 3$ and P_3 has no input and $\mathsf{P}_1, \mathsf{P}_2$ have no output.

We define security as in [Can00]. Here we give a few details for self containment. Each party P_i has a random tape r_i. In the pre-processing model or *correlated randomness model* $\boldsymbol{r} = (r_1, \ldots, r_n)$ is drawn from a joint distribution R,

$$(r_1, \ldots, r_n) \leftarrow R \ .$$

In the *standard* model, each r_i is uniform and independent of everything else.

We use

$$(y_1, \ldots, y_n) = \langle \mathsf{P}_1(x_1; r_1), \ldots, \mathsf{P}_n(x_n; r_n) \rangle$$

to denote a run of the protocol with input $\boldsymbol{x} = (x_1, \ldots, x_n)$ and fixed random tapes, resulting in $\mathsf{P}_i(x_i; r_i)$ outputting y_i. We use $c_{i,j}$ to denote the communication between $\mathsf{P}_i(x_i; r_i)$ and $\mathsf{P}_j(x_j; r_j)$. We let $c_{i,j} = c_{j,i}$. We let

$$\text{view}_i(\boldsymbol{x}, \boldsymbol{r}) = (x_i, r_i, c_{i,1}, \ldots, c_{i,n}, y_i) \ .$$

This is all the values seen by P_i in the protocol. In Fig. 1, the model is illustrated for $n = 3$ and for the case where P_3 has no input and $\mathsf{P}_1, \mathsf{P}_2$ have no output.

For a set $C \subseteq \{\mathsf{P}_1, \ldots, \mathsf{P}_n\}$ and an input vector $\boldsymbol{x}$ we let

$$\text{view}_C(\boldsymbol{x}, \boldsymbol{r}) = (\boldsymbol{x}, \{(i, \text{view}_i(\boldsymbol{x}, \boldsymbol{r}))\}_{i \in C}, \boldsymbol{y}) \ ,$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)$ and y_i is the output of P_i. We use $\text{view}_C\, \boldsymbol{x}$ to denote $\text{view}_C(\boldsymbol{x}, \boldsymbol{r})$ for a uniformly random $\boldsymbol{r}$

We now define perfect correctness and perfect privacy.

Definition 1 (perfect correctness). *For all inputs $(x_1, \ldots, x_n)$ and all random tapes $(r_1, \ldots, r_n)$ it holds that*

$$\langle \mathsf{P}_1(x_1; r_1), \ldots, \mathsf{P}_n(x_n; r_n) \rangle = f(x_1, \ldots, x_n) \ .$$

An adversary structure is a set $\mathcal{A}$ of subsets $C \subseteq \{\mathsf{P}_1, \ldots, \mathsf{P}_n\}$. It is usual to require that $\mathcal{A}$ is monotone but we do not do that here. For a simulator S and a set C of corrupted parties we define

$$\text{sim}_{C,S}\, \boldsymbol{x} = (\boldsymbol{x}, S\{(i, x_i, y_i)\}_{i \in C}, f\boldsymbol{x}) \ .$$

The simulator might be randomized, and we use $\text{sim}_{C,S}\, \boldsymbol{x}$ to denote the distribution obtained by a random run.

Definition 2 (perfect privacy). *We say that a protocol for f has perfect privacy against $\mathcal{A}$ if there exists a simulator S such that for all inputs $\boldsymbol{x}$ and $\boldsymbol{y} = f\boldsymbol{x}$ and all $C \in \mathcal{A}$ it holds that the distributions* $\text{sim}_{C,S}\, \boldsymbol{x}$ *and* $\text{view}_C\, \boldsymbol{x}$ *are the same.*

Note that perfect privacy implies perfect correctness.

When working with statistical security we introduce a security parameter $\sigma \in \mathbb{N}$. The protocol and the simulator is allowed to depend on σ. We use

$$(y_1, \ldots, y_n) = \langle \mathsf{P}_1(\sigma, x_1; r_1), \ldots, \mathsf{P}_n(\sigma, x_n; r_n) \rangle$$

to denote a run of the protocol with fixed security parameter σ and fixed random tapes, resulting in $\mathsf{P}_i(\sigma, x_i; r_i)$ outputting y_i. We let

$$\text{view}_i(\boldsymbol{x}, \boldsymbol{r}, \sigma) = (\sigma, x_i, r_i, c_{i,1}, \ldots, c_{i,n}, y_i) \ .$$

We use

$$(y_1, \ldots, y_n) \leftarrow \langle \mathsf{P}_1(\sigma, x_1), \ldots, \mathsf{P}_n(\sigma, x_n) \rangle$$

to denote a random run. In a random run, $\text{view}_i(\boldsymbol{x}, \sigma)$ becomes a random variable. For a simulator S, a set C of corrupted parties and security parameter σ we define

$$\text{sim}_{C,S}(\boldsymbol{x}, \sigma) = (\boldsymbol{x}, S(\{(i, x_i, y_i)\}_{i \in C}, \sigma), f\boldsymbol{x}) \ .$$

We use $\mathrm{D}(V_1, V_2)$ to denote the statistical distance between the distributions of random variables V_1 and V_2. Statistical security is defined as usual: even given the inputs and outputs of honest parties, the simulated views of the corrupted parties are statistically close to the real views.

Definition 3 (negligible function). *We call a function* $\epsilon : \mathbb{N} \to \mathbb{R}$ *negligible if for all* $c \in \mathbb{N}$ *there exists* $n \in \mathbb{N}$ *such that*

$$\forall n > n_0 \, (\, \epsilon(n) < n^{-c} \,) \ .$$

We use negl *to denote a generic negligible function, i.e., the term* negl *both takes the role as a function, but also has the implicit claim that this function is negligible.*

Definition 4 (statistical privacy). *We say that a protocol for* f *has statistical privacy against* $\mathcal{A}$ *if there exists a simulator* S *such that for all inputs* $\boldsymbol{x}$*, all values of* σ*,* $\boldsymbol{y} = f\boldsymbol{x}$*, and all* $C \in \mathcal{A}$ *it holds that*

$$\mathrm{D}(\mathrm{sim}_{C,S}(\boldsymbol{x}, \sigma), \mathrm{view}_C(\boldsymbol{x}, \sigma)) \ .$$

is negligible (as a function of σ*).*

We call a protocol t-private if it is private for the adversary set consisting of all subsets of size at most t.

2.3 Private Information Retrieval

A special case of MPC is private information retrieval. The setting is illustrated in Fig. 2. The input of P_1 is a bit string $x_1 \in \{0,1\}^I$. The input of P_2 specifies an index $x_2 \in \{0, \ldots, I-1\}$. The output y_2 is bit number x_2 in x_1. In the correlated randomness setting the randomness can be sampled as any joint distribution $(r_1, r_2) \leftarrow R$ and r_i securely given to P_i. We call this pre-processing PIR (PP-PIR). In contrast, PIR takes place in the standard model where r_1, r_2 are independent and uniform.

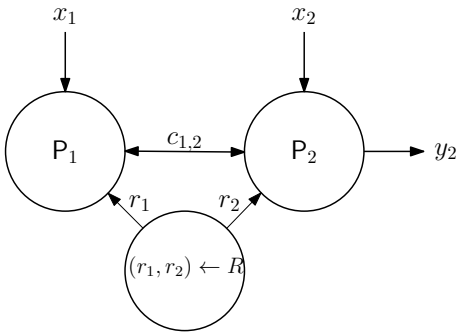

Fig. 2. PIR with pre-processing (PP-PIR).

Definition 5 (PIR). *We call π a perfect (PP-)PIR if it is perfectly correct and it is perfect $\{\{\mathsf{P}_1\}\}$-private, i.e., the view of P_i can be simulated given just x_1. We call π a statistical (PP-)PIR if it is statistical $\{\{\mathsf{P}_1\}\}$-private, i.e., the view of P_i can be simulated statistically given just x_1 and the protocol is statistically close to correct.*

We first (re)prove some basic facts about PIR. These results are known, at least in the folklore. However, we could not find a reference for the proof in the statistical security case, so we include proofs here for self-containment. Let $c_{1,2}$ denote the communication between P_1 and P_2. Then:

Lemma 3. *If π is a perfect PIR, then there exists a function x such that $x_1 = x(c_{1,2})$.*

Proof. The function postulated in the lemma can be implemented by computing each value $x_1[j]$ as follows: Given $c_{1,2}$, set $x_2 = j$ and iterate over all values of r_2, until one is found where where $(x_2, r_2, c_{1,2})$ is a possible value of P_2's view of π. More concretely, if P_2 starts from x_2, r_2 and we assume P_1 sent the messages in $c_{1,2}$ (with P_1 as sender), then P_2 would send the messages occurring in $c_{1,2}$ (with P_2 as sender). Once such an r_2 is found, output the value y that P_1 would output based on this view. It now follows immediately from perfect correctness that if the loop terminates, then $y = x_1[j]$. Moreover, perfect privacy implies that an r_2 as required for termination must exist: Given any view $x_1, r_1, c_{1,2}$ for P_1, then for any x_2 there must be an r_2 leading to this view. Otherwise, P_1 could exclude one or more values of x_2. □

Lemma 4. *Assume that π is a statistical PIR. Let X_1, X_2 denote random variables describing uniformly random inputs to $\mathsf{P}_1, \mathsf{P}_2$. Let $C_{1,2}$ be the random variable describing $c_{1,2}$ after a random run on X_1, X_2. Then there exists a function x such that $\Pr[X_1 = x(C_{1,2})] = 1 - \mathrm{negl}(\sigma)$.*

Proof. Let $C_{1,2}(x_2)$ denote $C_{1,2}$ when the input of P_2 is x_2. We now prove two claims.

Claim 1. There exists a function x_{x_2} such that

$$\Pr[X_1[x_2] = x_{x_2}(C_{1,2}(x_2))] = 1 - \mathrm{negl}(\sigma) \ .$$

Claim 2. For all x_2 and x_2' it holds that

$$\mathrm{D}((X_1, C_{1,2}(x_2)), (X_1, C_{1,2}(x_2'))) = \mathrm{negl}(\sigma) \ .$$

Let us first see that if these claims are true, then we are done. By combining the claims we get that:

$$\Pr[X_1[x_2] = x_{x_2}(C_{1,2}(x_2'))] = 1 - \mathrm{negl}(\sigma) \ .$$

Now let $x(C) = (x_0(C), \ldots, x_{I-1}(C))$. Then by a union bound

$$\Pr[X_1 = x(C_{1,2}(x_2'))] = 1 - \mathrm{negl}(\sigma) \ ,$$

as I is polynomial in σ. This holds for all x_2', so

$$\Pr[X_1 = x(C_{1,2})] = 1 - \mathrm{negl}(\sigma) \ ,$$

as desired.

Claim 1 follows from statistical correctness. Consider a random run of P_2 using input x_2 and uniformly random (r_1, r_2), resulting in communication $c_{1,2}$ and output y_2. We know that

$$\Pr\left[y_2 = X_1[x_2]\right] = 1 - \mathrm{negl}(\sigma) \ .$$

Assume now that someone gave you the execution of the protocol but deleted x_1, r_1, r_2, and y_2, and hence left you with only $c_{1,2}$ and x_2. Consider now sampling a uniformly random x_1', r_1' and r_2' that are consistent with $c_{1,2}$ and x_2, i.e., running $\mathsf{P}_1(x_1'; r_1')$ and $\mathsf{P}_2(x_2; r_2')$ produced exactly the messages $c_{1,2}$. Let y_2' be the resulting output of $\mathsf{P}_2(x_2; r_2')$ when running $\mathsf{P}_1(x_1'; r_1')$ and $\mathsf{P}_2(x_2; r_2')$.

Then clearly y_2' and y_2 will have the same distribution. Namely, the distribution of the deleted x_1, r_1 and r_2 were also uniform, consistent with $c_{1,2}, x_2$. Hence

$$\Pr\left[y_2' = X_1[x_2]\right] = 1 - \mathrm{negl}(\sigma) \ .$$

Let y be the function which samples y_2' from $c_{1,2}, x_2$ as described above. Let $x_{x_2}(\cdot) = y(\cdot, x_2)$. Then

$$\Pr\left[x_{x_2}(C_{1,2}(x_2)) = X_1[x_2]\right] = 1 - \mathrm{negl}(\sigma) \ ,$$

as desired.

Claim 2 follows directly from statistical privacy (P_1 does not learn x_2). Namely, we have that

$$\mathrm{sim}_{\{\mathsf{P}_1\},S}(\boldsymbol{x}, \sigma) = ((X_1, x_2), S(X_1, \sigma), X_1[x_2])$$

and

$$\mathrm{view}_{\{\mathsf{P}_1\}}(\boldsymbol{x}, \sigma) = ((X_1, x_2), (X_1, C_{1,2}), X_1[x_2])$$

are statistically indistinguishable, so if we let $C_{1,2}'$ be the distribution of $C_{1,2}$ output by S, then

$$\mathrm{D}((X_1, C_{1,2}(x_2)), (X_1, C_{1,2}')) = \mathrm{negl}(\sigma)$$

for all x_2. Then use the triangle inequality:

$$\mathrm{D}((X_1, C_{1,2}(x_2)), (X_1, C_{1,2}(x_2'))) \leq$$
$$\mathrm{D}((X_1, C_{1,2}(x_2)), (X_1, C_{1,2}')) + \mathrm{D}((X_1, C_{1,2}'), (X_1, C_{1,2}(x_2'))) = \mathrm{negl}(\sigma).$$

□

These results imply that the communication in single server PIR must be large: By Lemma 4 and Lemma 1 we can conclude that $H(C_{1,2}) \geq I(X_1; C_{1,2}) = H(X_1) - H(X_1|C_{1,2}) \geq H(X_1) - \text{negl}(\sigma)$. We now show that a similar result holds for PP-PIR:

Lemma 5. *Assume that π is a statistical PP-PIR. Let X_1, X_2 denote random variables describing uniformly random inputs to $\mathsf{P}_1, \mathsf{P}_2$. Let $C_{1,2}$ be the random variable describing $c_{1,2}$ after a random run on X_1, X_2. Then $H(C_{1,2}) \geq H(X_1) - \text{negl}(\sigma)$.*

Proof. Let R be the function used to sample the correlated randomness (r_1, r_2), i.e., $(r_1, r_2) = R(r)$ for a uniformly random r. Notice that since (PP-)PIR does not impose any privacy restrictions on what P_2 learns, we can construct a secure PIR protocol π' from π as follows: P_2 runs R, sends r_1 to P_1, and then we run π. We can now apply Lemma 4 and Lemma 1 to π' and conclude that $H(X_1|C_{1,2}, R_1) = \text{negl}(\sigma)$, here R_1 is a random variable describing the choice of r_1 and we note that the conversation in π' consists of r_1 and $c_{1,2}$. Since X_1 and R_1 are independent, we have $H(X_1) = H(X_1|R_1)$ and now the chain rule gives immediately that $H(C_{1,2}) \geq H(X_1) - \text{negl}(\sigma)$ as desired (intuitively, given R_1, the uncertainty on X_1 is maximal, but if we add $C_{1,2}$ the uncertainty drops to essentially 0, and so $C_{1,2}$ must contain information corresponding to the entropy drop). □

3 Lower Bounds Without Correlated Randomness

In this section we prove that there is an n-party function describable by a circuit C of size $|C|$ where each of the n parties have communication $\Theta(|C|)$, in the standard model. For the sake of presentation we present it via a series of simpler results, each highlighting one essential idea of the proof. We first give a function for three parties where one party must have high communication, proving the result first for perfect and then statistical security. Then we lift this up to an n-party function where there is a special *heavy* party. A heavy party has a short input and a short output, but still must have communication $\Theta(|C|)$ bits. Then we embed this function into a slightly more complicated one, where each party can obliviously choose to be the heavy party. This gives an n-party function where all parties must have communication $\Theta(|C|)$. This is because they must have communication $\Theta(|C|)$ when they are the heavy party, and a private protocol is not allowed to leak who is the heavy party. Throughout this series of results we assume maximal threshold $n = 2t + 1$ for simplicity. At the end we investigate how the bound behaves when $n = 2t + s$ for $1 \leq s \leq t$.

Our main theorem will be the following.

Theorem 1. *Let $n = 2t+s$. There exists a function $\widehat{\text{IP}}_{I,n}$ with circuit complexity $O(nI)$ such that in any statistically t-private protocol for $\widehat{\text{IP}}_{I,n}$ in the model without preprocessing, the average communication complexity is at least $\frac{Int}{2s} - \epsilon = \Theta(ntI)/s$ for a negligible ϵ.*

3.1 Lower Bound, Perfect Security, Three Parties

We start by considering a protocol for three parties of the form in Fig. 1. The input of P_1 is $x_1 \in \{0,1\}^I$, the input of P_2 is $x_2 \in \{0,1\}^I$. The output of P_3 is the inner product between x_1 and x_2, i.e., the single bit

$$y_3 = \bigoplus_{i=1}^{I} x_{1,i} x_{2,i} \ .$$

Denote this function by $\mathrm{IP}_{I,3}$.

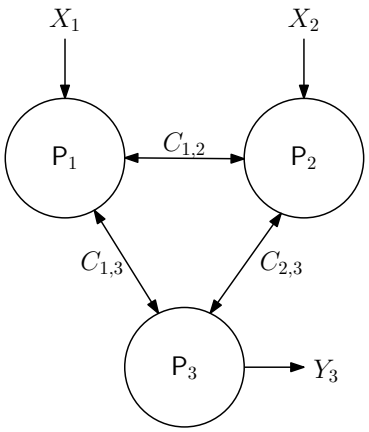

Fig. 3. A special case of the model where $n = 3$ and P_3 has no input and $\mathsf{P}_1, \mathsf{P}_2$ have no output, and where the inputs are uniformly random.

Theorem 2. *In any protocol for* $\mathrm{IP}_{I,3}$ *that is perfectly correct and perfectly private if* P_1 *or* P_2 *are corrupt, party* P_3 *will for random inputs have average communication complexity at least* I*.*

Proof. Assume that we have a protocol implementing $\mathrm{IP}_{I,3}$ with security as assumed. Let X_1 denote a random variable that is uniformly random on $\{0,1\}^I$. Let X_2 denote an independent random variable that is uniformly random on $\{0,1\}^I$. Let $C_{i,j}$ denote the communication between P_i and P_j in a random execution $\langle \mathsf{P}_1(X_1), \mathsf{P}_2(X_2), \mathsf{P}_3 \rangle$ and let Y_3 denote output of P_3 in the random execution. See Fig. 3.

Below, we will first prove that the following two inequalities implies high communication for P_3:

$$\mathrm{H}(X_1 \mid C_{1,2}, C_{1,3}, C_{2,3}) \leq \epsilon \ . \tag{1}$$

$$\mathrm{H}(X_1 \mid C_{1,2}, C_{2,3}) \geq I - \epsilon \ . \tag{2}$$

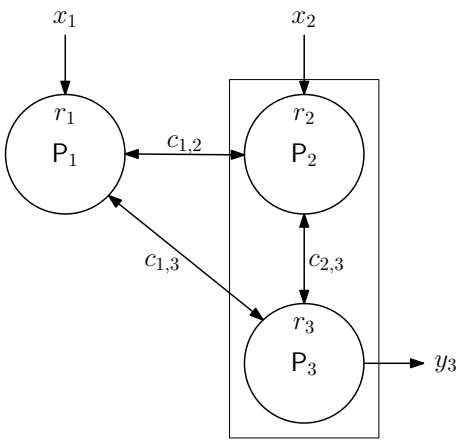

Fig. 4. Collapsing P_2 and P_3 into a single party.

These inequalities will be true with $\epsilon = 0$ for perfect security and for a negligible ϵ for statistical security. We will show that this implies:

$$\mathrm{H}(C_{1,3}) \geq I - 2\epsilon \ , \tag{3}$$

To see this, we use the chain rule for conditional Shannon entropy:

$$I - \epsilon \leq H(X_1 \mid C_{1,2}, C_{2,3}) \leq H(X_1 C_{1,3} \mid C_{1,2} C_{2,3}) =$$
$$H(X_1 \mid C_{1,3} C_{1,2} C_{2,3}) + H(C_{1,3} \mid C_{1,2} C_{2,3}) \leq \epsilon + H(C_{1,3}).$$

We conclude that $H(C_{1,3}) \geq I - 2\epsilon$, i.e., P_3 must communicate on average at least $1 - 2\epsilon$ bits.

We now prove that for a perfectly secure protocol, (1) holds with $\epsilon = 0$. For this purpose, consider the 3-party protocol π' in Fig. 4, where we consider P_2 and P_3 as one party. We call P_1 the sender and $(\mathsf{P}_2, \mathsf{P}_3)$ the receiver. Notice that x_2 can be taken to be any vector which is all-zero, except it has a 1 in position j. In that case it follows from perfect correctness of π that the receiver always learns the j'th bit of x_1. Furthermore, if π is perfectly private when P_1 is corrupted, then the sender learns nothing about j. This is because a corrupted sender learns only x_1 and r_1, exactly as in the protocol. So, if π is a perfectly correct and perfectly 1-private protocol for $\mathrm{IP}_{3,I}$, then π' is a perfect PIR. Hence (1) follows from Lemma 3.

We then prove (2) for $\epsilon = 0$. To see this note that by perfect privacy when P_2 is corrupt, we can simulate $(C_{1,2}, C_{2,3})$ given X_2 as P_2 has no output. This implies that

$$\mathrm{H}(X_1 \mid C_{1,2}, C_{2,3}) \geq \mathrm{H}(X_1 \mid X_2) = I$$

as we wanted.

This completes the proof of Theorem 2. □

3.2 Lower Bound, Statistical Security, Three Parties

We now prove that essentially the same result holds also for statistical security.

Theorem 3. *In any protocol for* $\mathrm{IP}_{I,3}$ *that is statistically correct and statistically private if* P_1 *or* P_2 *are corrupt, party* P_3 *will for random inputs have average communication complexity at least* $I - \epsilon$ *for a negligible* ϵ*.*

Proof. From the previous section it is clear that we only have to prove that (1) and (2) still hold.

As for (1), we clearly get a statistically secure PIR by considering P_2 and P_3 as one party, exactly as in the proof for perfect security. Then, by Lemma 4, it follows that given $C_{1,2}, C_{1,3}$ one can compute a guess at X_1' such that $\Pr[X_1' \neq X_1]$ is negligible. Then (1) follows by Lemma 1:

$$\mathrm{H}(X|C_{1,2}, C_{1,3}, C_{2,3}) \leq \mathrm{H}(X \mid C_{1,2}, C_{1,3}) \leq \mathrm{H}(X_1 \mid X_1') \leq \epsilon$$

for a negligible ϵ.

As for (2), we exploit the fact that the protocol is statistically secure against a corrupt P_2. This means there exists a simulator that (using only x_2 as input) will simulate the view of P_2, including $c_{1,2}, c_{2,3}$. The definition of statistical security requires that the simulated view is statistically close to the real view even given the input x_1 (of the honest P_1). Note that here the distributions are taken only over the random coins of the parties and the simulator.

Now we run the protocol with a uniformly random X_1 as input for P_1, and a uniformly random input X_2 for P_2. As before we let $C_{1,2}, C_{2,3}$ denote the variables representing the communication in the real protocol while $C_{1,2}', C_{2,3}'$ denote the simulated conversation. The statistical security now implies that

$$\mathrm{D}((X_1, (C_{1,2}, C_{2,3})),\ (X_1, (C_{1,2}', C_{2,3}')))$$

is negligible — actually statistical security implies the stronger requirement that the distance be small for every fixed value of X_1 and X_2. Now (2) follows immediately from this and Corollary 1.

This completes the proof of Theorem 3. □

3.3 Lower Bound, Statistical Security, n Parties, Maximal Resilience

We now generalize the bound to more parties. Assume that $n = 2t + 1$. We will call the parties $\mathsf{P}_{1,1}, \ldots, \mathsf{P}_{1,t}, \mathsf{P}_{2,1}, \ldots, \mathsf{P}_{2,t}, \mathsf{P}_3$. We assume that P_3 only has output and the other parties only have inputs. Consider the following function $\mathrm{IP}_{n,I}$, where each $\mathsf{P}_{j,i}$ for $i = 1, \ldots, t; j = 1, 2$ has an input $x_{j,i} \in \{0,1\}^I$ and no output, and where P_3 has no input and has an output $y_n \in \{0,1\}$. The output y_n is the inner product between $x_{1,1}x_{1,2}\ldots, x_{1,t}$ and $x_{2,1}x_{2,2}\ldots, x_{2,t}$ computed in the field with two elements. See Fig. 5.

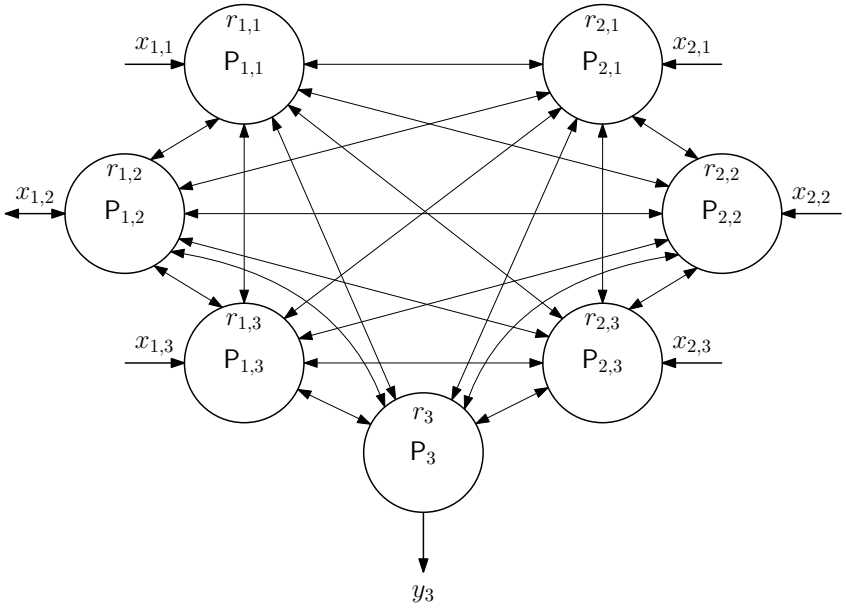

Fig. 5. A special case of the model where $n = 7$ and P_3 has no input and $\mathsf{P}_{1,1}, \mathsf{P}_{1,2}, \mathsf{P}_{1,3}, \mathsf{P}_{2,1}, \mathsf{P}_{2,2}, \mathsf{P}_{2,3}$ have no outputs.

Theorem 4. *Let $n = 2t + 1$. In any statistically t-private and statistically correct protocol for $\mathrm{IP}_{I,n}$ party P_3 will for all inputs have average communication complexity at least $tI - \epsilon$ for a negligible ϵ.*

Proof. Given a protocol for $\mathrm{IP}_{I,n}$, we can make a protocol for $\mathrm{IP}_{tI,3}$ by grouping parties together as in Fig. 6. Corrupting one party in $\mathrm{IP}_{tI,3}$ corrupts at most t parties in $\mathrm{IP}_{I,n}$. Therefore we can apply Theorem 3. □

3.4 Stronger Lower Bound, Statistical Security, n Parties, Maximal Threshold

We now give a function where all parties need to have high communication complexity. We do this essentially by making a function where each party obliviously can choose to be the party P_3 in the proof of Theorem 4. Since nobody knows who plays the role of P_3 and P_3 needs to have high communication complexity, all parties must have high communication complexity.

Assume that $n = 2t+1$. We will call the parties $\mathsf{P}_{1,1}, \ldots, \mathsf{P}_{1,t}, \mathsf{P}_{2,1}, \ldots, \mathsf{P}_{2,t}, \mathsf{P}_3$. Consider the following function $\mathrm{IP}'_{n,I}$, where each $\mathsf{P}_{j,i}$ for $i = 1, \ldots, t; j = 1, 2$ has an input $x_{j,i} \in \{0,1\}^I$ and an input $b_{j,i} \in \{0,1\}$, and where P_3 has input $b_3 \in \{0,1\}$. First compute y to be the inner product between $x_{1,1}x_{1,2}\ldots, x_{1,t}$ and $x_{2,1}x_{2,2}\ldots, x_{2,t}$. The output of P_3 is $y_3 = b_3 y$. The output of $\mathsf{P}_{j,i}$ is $y_{j,i} = b_{j,i} y$.

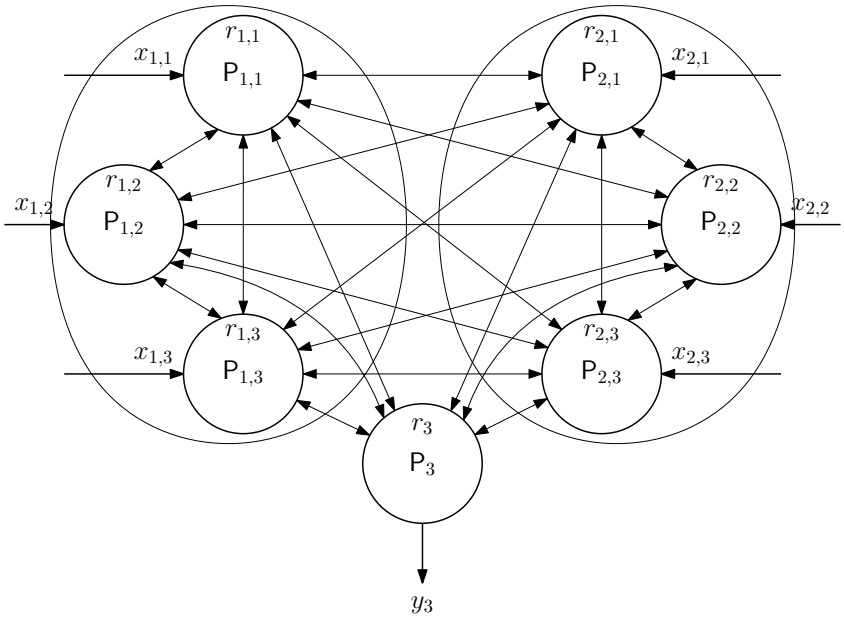

Fig. 6. Reduction from the n-party case to the 3-party case, maximal threshold $n = 2t + 1$.

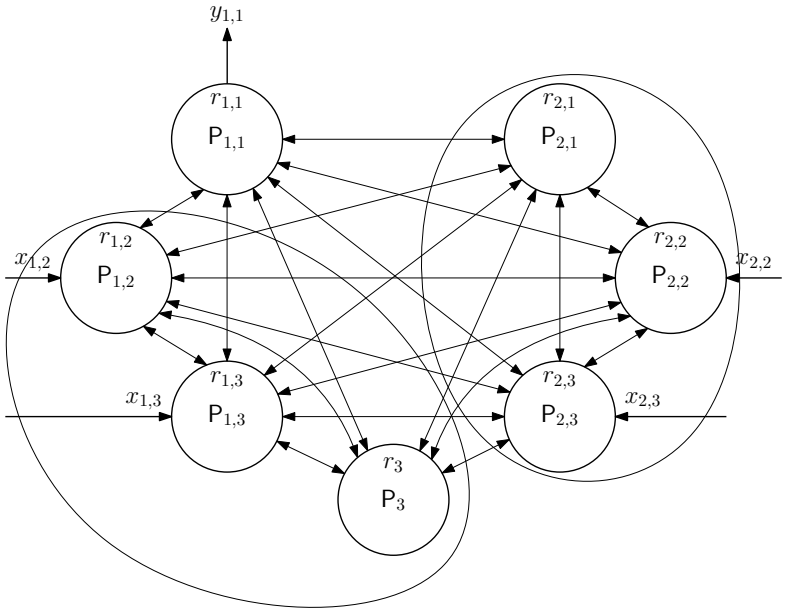

Fig. 7. Reduction from IP to IP′.

Theorem 5. *Let $n = 2t+1$. In any statistically t-private and statistically correct protocol for $\mathrm{IP}'_{I,n}$ the average total communication is at least $(n(t-1)I)/2 - \epsilon$ for a negligible ϵ.*

Proof. Assume we have such a protocol for $\mathrm{IP}'_{I,n}$. Notice that if we pick any input except that we hard-code the inputs $b_3 = 1$ and $b_{j,i} = 0$, then $\mathrm{IP}'_{I,n}$ is just $\mathrm{IP}_{I,n}$, so it follows trivially that for these inputs the communication complexity of P_3 is $tI - \epsilon$. And this holds for all possible inputs (by statistical security and by considering the case where no parties are corrupted), in particular also the inputs where we set all non-hardcoded inputs to be all-zero, i.e., $x_{j,i} = \mathbf{0}$ and $x_3 = \mathbf{0}$. Call this input vector $\boldsymbol{x}_3$.

Consider then hard-coded inputs where we make the change that $b_3 = 0$, $b_{1,1} = 1$, $b_{j,i} = 0$ for $(j,i) \neq (1,1)$, $x_{1,1} = \mathbf{0}$, and $x_{2,1} = \mathbf{0}$. If we have a secure protocol for $\mathrm{IP}'_{n,I}$ we of course also have one for the case with these hard-coded inputs. We can then via the reduction in Fig. 7 apply Theorem 3 to see that the communication complexity of $\mathsf{P}_{1,1}$ must be at least $(t-1)I - \epsilon$. Note that it is $t-1$ and not t as we had to get rid of the input of $\mathsf{P}_{1,1}$ to be able to reduce to the three-party case. The communication complexity of $\mathsf{P}_{1,1}$ is at least $(t-1)I - \epsilon$ for all ways to set the non-hardcoded inputs, so also when we set them to be all-zero. Call this input vector $\boldsymbol{x}_{1,1}$.

Similarly, define $\boldsymbol{x}_{j,i}$ to be the set of inputs where all inputs are 0 except that $b_{j,i} = 1$. We can conclude as above, that on this input $\mathsf{P}_{j,i}$ has communication complexity at least $(t-1)I - \epsilon$.

Consider then the input vector $\mathbf{0}$ where all inputs are 0. The only difference between for instance $\boldsymbol{x}_{j,i}$ and $\mathbf{0}$ is whether $b_{j,i} = 1$ or $b_{j,i} = 0$. Notice, however, that since all other inputs are 0, this change does not affect the output of any other party. Therefore their views cannot change by more than a negligible amount. This easily implies that the average amount of communication with $\mathsf{P}_{j,i}$ cannot change by more than a negligible amount. By linearity of expectation it follows that the average communication complexity of $\mathsf{P}_{j,i}$ cannot change by more than a negligible amount. So on input $\mathbf{0}$ party $\mathsf{P}_{j,i}$ will have average communication complexity negligibly close to $(t-I)I - \epsilon$. This holds for all parties. Therefore the average total communication is at least $(n(t-1)I)/2 - \epsilon/2$. It is not $(t-1)I$ as we would be counting each bit of communication twice (both at the sending and the receiving end). □

3.5 Lower Bound, Statistical Security, n Parties, Sub-Maximal Resilience

We now generalize our bound to the case with sub-maximal threshold, i.e., $n > 2t + 1$. Let $s = n - 2t$. We will first show that one group of s players must communicate a lot. We consider the function $\mathrm{IP}_{I,n,t}$, where each $\mathsf{P}_{j,i}$ for $i = 1, \ldots, t; j = 1, 2$ has an input $x_{j,i} \in \{0,1\}^I$ and no output, and where $\mathsf{P}_{3,1}, \ldots, \mathsf{P}_{3,s}$ have no input, and $\mathsf{P}_{3,1}$ has an output $y_n \in \{0,1\}$ which is the inner product of between $x_{1,1}x_{1,2}\ldots, x_{1,t}$ and $x_{2,1}x_{2,2}\ldots, x_{2,t}$ computed in the field with two elements. Call this function $\mathrm{IP}_{I,n,t}$. See Fig. 8.

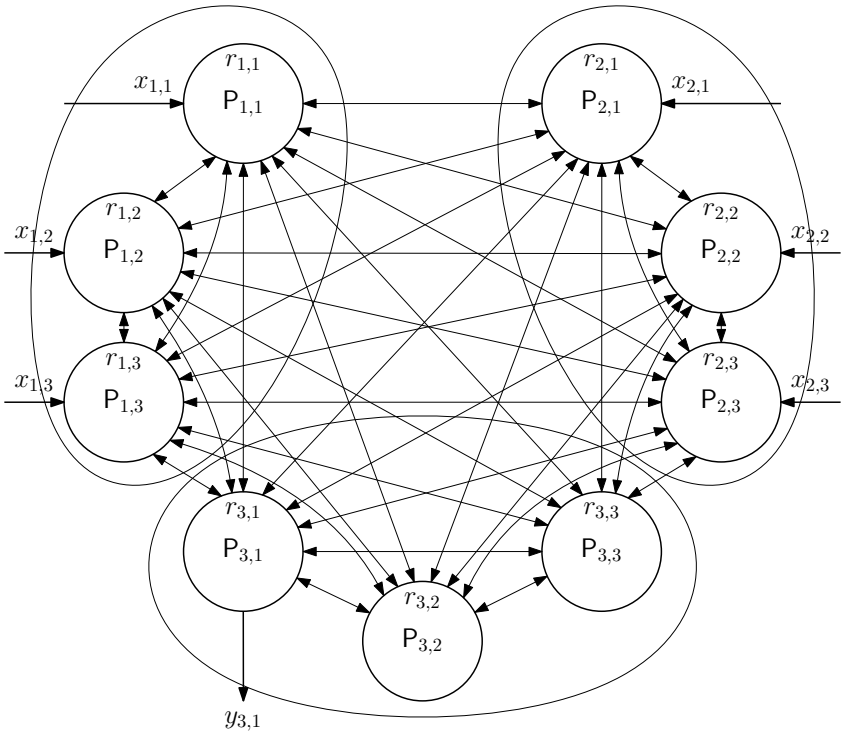

Fig. 8. Reduction from the n-party case to the 3-party case, sub-maximal threshold, here $n = 9$ and $t = 3$.

Theorem 6. *Let $s = n - 2t$. In any statistically t-private protocol for $\mathrm{IP}_{I,n,t}$ parties $\mathsf{P}_{3,1}, \ldots, \mathsf{P}_{3,s}$ will for all inputs have average total communication complexity at least $tI - \epsilon$ for a negligible ϵ.*

Proof. Given a protocol for $\mathrm{IP}_{I,n,t}$, we can make a 3-party protocol for $\mathrm{IP}_{tI,3}$ by grouping parties together as in Fig. 8. This protocol is secure against corruption of P_1 or P_2 since this corrupts at most t parties in the protocol for $\mathrm{IP}_{n,I,t}$. Therefore we can apply Theorem 3 (recall that to show that result, we only needed to corrupt P_1 or P_2). □

3.6 Stronger Lower Bound, Statistical Security, n Parties, Sub-Maximal Threshold

Assume that $n = 2t + s$. Assume for simplicity that s is even and that s divides n. Let $n = 2T$.

We will call the parties $\mathsf{P}_{1,1}, \ldots, \mathsf{P}_{1,T}, \mathsf{P}_{2,1}, \ldots, \mathsf{P}_{2,T}$. Consider the function $\widehat{\mathrm{IP}}_{n,I}$. Each $\mathsf{P}_{j,i}$ for $i = 1, \ldots, t; j = 1, 2$ has an input $x_{j,i} \in \{0,1\}^I$ along with an input $b_{j,i} \in \{0,1\}$ and an output $y_{j,i} \in \{0,1\}$. The outputs are defined as follows. First let y be the inner product between $x_{1,1}x_{1,2}\ldots, x_{1,T}$ and $x_{2,1}x_{2,2}\ldots, x_{2,T}$ computed in the field with two elements. Let $y_{j,i} = b_{j,i}y$.

We prove Theorem 1, which we recall here:

Theorem 7. *Let $n = 2t+s$. There exists a function $\widehat{\mathrm{IP}}_{I,n}$ with circuit complexity $O(nI)$ such that in any statistically t-private protocol for $\widehat{\mathrm{IP}}_{I,n}$ in the model without preprocessing, the average communication complexity is at least $\frac{Int}{2s} - \epsilon = \Theta(ntI)/s$ for a negligible ϵ.*

Proof. Assume we have a protocol for $\widehat{\mathrm{IP}}_{I,n}$. Let $h = s/2$. We can group the parties into n/s groups of s parties, indexed by $g = 0, \ldots, n/s - 1$. In group G_g we put the parties $\mathsf{P}_{1,hg+1}, \ldots, \mathsf{P}_{1,hg+h}$ and $\mathsf{P}_{2,hg+1}, \ldots, \mathsf{P}_{2,hg+h}$.

For each g we can define three virtual parties $\mathsf{P}_1^g, \mathsf{P}_2^g, \mathsf{P}_3^g$. We let $\mathsf{P}_3^g = G_g$. We let $\mathsf{P}_1^g = \{\mathsf{P}_{1,1}, \ldots, \mathsf{P}_{1,T}\} \setminus G_g$ and we let $\mathsf{P}_2^g = \{\mathsf{P}_{2,1}, \ldots, \mathsf{P}_{2,T}\} \setminus G_g$. We then hardcode the inputs of the parties in G_g to be all-zero, except that we let $\mathsf{P}_{1,hg+1}$ choose to be the heavy party by setting $b_{1,hg+1} = 1$. For all other parties, let them use $b_{j,i} = 0$. It follows by statistical security, as in the proof of Theorem 5, that the communication complexity for these hardcoded inputs must be the same as for some fixed input, say the all-0 one.

Note that $|\mathsf{P}_1^g| = |\mathsf{P}_2^g| = T - s/2 = t$. So if the protocol we start from is private against t corruptions, then the derived protocol for the three virtual parties is private against corruption of P_1^g or P_2^g. By Theorem 3, it follows that P_3^g must communicate at least $tI - \epsilon$ bits. There are n/s groups. Since the choice of g depends only on the private inputs, we can argue exactly as in the proof of Theorem 5 that all groups must communicate this much, so this gives a total communication of at least $(tIn/s)/2 - \epsilon/2$.

Finally, it is easy to see that the circuit complexity of $\widehat{\mathrm{IP}}_{n,I}$ is $O(nI)$, since the cost of computing the function is dominated by the cost of computing the inner product. □

4 Lower Bounds, Correlated Randomness

In this section, we consider lower bounds for protocols in the correlated randomness model and arrive at the following result:

Theorem 8. *Let $n = t + s$. There exists a function $PIR_{n,I}$ with circuit complexity $O(nI)$ such that in any statistically t-private protocol for $PIR_{n,I}$ in the preprocessing model, the average communication complexity is at least $\Theta(ntI)/s$.*

We sketch the proof of this result, the details are trivial to fill in, as they are extremely similar to the ideas in the previous section.

We define the function $PIR_{n,I}$ as follows: each party P_i has three inputs: $x_i \in \{0,1\}^I$, $z_i \in \{0,1\}^{\lg(nI)}$ and $b_i \in \{0,1\}$. To evaluate the function, set x to be the concatenation of all the x_i's and set $z = \oplus_{i=1}^n z_i$. Interpret z as an index that points to a bit in x which we denote $x[z]$. Then the output for P_i is $b_i \cdot x[z]$.

Assume first that we have a protocol π that computes $PIR_{I,n}$ with statistical security in the correlated randomness model when $t = n-1$ parties are corrupted. We consider the case $s > 1$ later.

For any fixed value $1 \leq i \leq n$, we can group the parties $\{\mathsf{P}_j |\ j \neq i\}$ together to form one virtual party P_i^1, and let P_i play the role of a second virtual party P_i^2. Furthermore we hardcode the inputs as follows: $b_i = 1$ and $b_j = 0$ for $j \neq i$, and furthermore $z_j = 0^{\lg(nI)}$ for $j \neq i$. With this hardcoding we clearly obtain a PP-PIR where P_i^1 is the sender and P_i^2 is the receiver. It follows from Lemma 5 that the communication complexity for P_i^2 must be $\Omega(nI)$. Since this holds for any i, and since the communication pattern is not allowed to depend on the inputs, it follows as in the proof of Theorem 5 that all players must have this much communication always, so we see that the total communication complexity is $\Omega(n^2 I)$.

Assume now that the threshold t is sub-optimal, i.e., $t = n - s$, where we assume for simplicity that s divides n. Now, given a protocol that computes $PIR_{I,n}$ in this setting, we can divide the set of players in n/s disjoint subsets of size s and show that each group of s players must have communication complexity $\Omega(nI)$. This follows similarly to what we just did, by hardcoding the inputs appropriately. As a result we get a lower bound of $\Omega(ntI/s)$ for this case.

Finally, we note that for any all large enough I (compared to n), the circuit complexity of $PIR_{n,I}$ is $O(nI)$. To see this, note that the cost of computing the function is dominated by computing $x[z]$ from x, z. This is known as the storage access function and is known to have a linear size circuit [Weg87].

5 Upper bounds

5.1 Honest majority

In this section, we prove upper bounds that match up to a constant factor the lower bounds we proved for the standard model with honest majority. At first sight this may seem like a trivial exercise: In [DN07] a passively secure protocol

was presented that securely evaluates any arithmetic circuit C of size $|C|$ with communication complexity $O(n|C|)$ field elements. This seems to already match our lower bound. However, that protocol only works for a field $\mathbb{F}$ with more than n elements, and so cannot be directly used for the Boolean case.

One can partially resolve this by noticing that all our lower bounds hold for any finite field, in fact the proofs do not use the size of field at all. So if we consider instead the inner product function over a larger field $\mathbb{F}$, then the bounds match. But this is still not completely satisfactory because the result still holds only as long as $n < |\mathbb{F}|$.

To get a cleaner result, we can combine the protocol from [DN07] with a recent technique from [CCXY18] known as *reverse multiplication friendly embeddings* (RMFE). Such an embedding can be defined when we have a base field $\mathbb{F}$ and an extension field $\mathbb{K}$. Then the embedding consists of two $\mathbb{F}$-linear mappings S, T where $S : \mathbb{F}^k \mapsto \mathbb{K}$ and $T : \mathbb{K} \mapsto \mathbb{F}^k$. The defining property we need is that

$$T(S(\boldsymbol{a}) \cdot S(\boldsymbol{b})) = \boldsymbol{a} * \boldsymbol{b}$$

for any $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}^k$, and where $\boldsymbol{a} * \boldsymbol{b}$ is the coordinate-wise (Schur) product.

So these mappings allow us to implement k multiplications in parallel in $\mathbb{F}$ by one multiplication in K. In [CCXY18] it is shown how to construct (families of) RMFE(s) such that $\mathbb{F} = \mathbb{F}_2$ and $\mathbb{K} = \mathbb{F}_{2^u}$ where u is $\Theta(k)$. So the encoding of $\boldsymbol{a}$ as an element in $\mathbb{K}$ comes with only a constant factor overhead. With these tools, we can prove:

Theorem 9. *There exists a perfect passive secure protocol for honest majority such that for any n and all large enough I, the protocol computes $IP'_{I,n}$ with communication complexity $O(n^2 I)$ bits.*

Remark 1. Since the protocol handles $n = 2t + 1$ this matches our upper bound in Theorem 5, up to a constant factor.

Proof. (Sketch) First we choose an RMFE by the above construction, so we have $S : \mathbb{F}^k \mapsto \mathbb{K}$ and $T : \mathbb{K} \mapsto \mathbb{F}^k$, we make the choice such that $n < |\mathbb{K}| = 2^u$. Then the protocol we build will work as long as $I \geq k$.

Recall that in the function $IP'_{I,n}$, which is defined at the start of Section 3.4, the first $2t$ parties get as input a vector consisting of I bits. We will call these the *vector parties*. In addition, each party also gets an input bit that decides if that party gets output. For convenience in this proof, we will denote the parties by a single index, so that P_j, for $j = 1..2t$ are the input parties, whereas P_n's only input is the bit b_n. Initially, each vector party will split his input vector into $\lceil I/k \rceil$ vectors of length k bits each, padding the last block with 0's if it is incomplete. By appropriate renumbering we can say that between them, the vector parties now hold k-bit vectors $\boldsymbol{x}_1, ..., \boldsymbol{x}_v$ and $\boldsymbol{y}_1, ..., \boldsymbol{y}_v$, where party P_j holds a subset of the $\boldsymbol{x}_i$'s if $1 \leq j \leq t$, and holds a subset of the $\boldsymbol{y}_i$'s if $t < j \leq 2t$. Let $\boldsymbol{x}$ be the concatenation of all the $\boldsymbol{x}_i$'s and $\boldsymbol{y}$ the concatenation of all $\boldsymbol{y}_i$'s. Now the desired output for party P_j, for all j, can be written as $b_j(\boldsymbol{x} \cdot \boldsymbol{y})$ where $\boldsymbol{x} \cdot \boldsymbol{y}$ is the inner product.

Now, note that one way to compute $\boldsymbol{x} \cdot \boldsymbol{y}$ product is to first compute $\boldsymbol{z} = \sum_i \boldsymbol{x}_i * \boldsymbol{y}_i$ and then add all coordinates in $\boldsymbol{z}$ (recall that $*$ denotes the Schur or coordinate-wise product). This is essentially the strategy we will follow.

Recall that each vector party P_j holds a subset of $\boldsymbol{x}_i$'s or a subset of $\boldsymbol{y}_i$'s. He applies S to each vector in his subset to get a set V_j of elements in $\mathbb{K}$. The parties will now use the V_j's as input to an instance of the protocol from [DN07]. This protocol can compute any arithmetic circuit over $\mathbb{K}$ and is based on Shamir secret sharing. It can therefore be used to compute securely $[\sum_i S(\boldsymbol{x}_i) \cdot S(\boldsymbol{y}_i)]$, which denotes a secret sharing of $\sum_i S(\boldsymbol{x}_i) \cdot S(\boldsymbol{y}_i)$, i.e., each party holds a share of the value.

Let $w = \sum_i S(\boldsymbol{x}_i) \cdot S(\boldsymbol{y}_i)$. Note that by linearity

$$T(w) = T(\sum_i S(\boldsymbol{x}_i) \cdot S(\boldsymbol{y}_i)) = \sum_i T(S(\boldsymbol{x}_i) \cdot S(\boldsymbol{y}_i)) = \sum_i \boldsymbol{x}_i * \boldsymbol{y}_i = \boldsymbol{z}$$

So this means that the only remaining problem is the following: given a secret sharing of w, we need to securely compute $T(w)$ and add all coordinates of the resulting vector. The result of this will be $\boldsymbol{x} \cdot \boldsymbol{y}$, the result we want. If we think of $\mathbb{K}$ as a u-dimensional vector space over $\mathbb{F}$, the combined operation of applying T and adding the coordinates is an $\mathbb{F}$-linear mapping and hence has a matrix M over $\mathbb{F}$, actually with just 1 row. Therefore we will first compute sharings $[w_1], ..., [w_u]$ where he w_i's are the coordinates of w. This can be done by a standard method where we first create $[r], [r_1], ..., [r_u]$ for a random $r \in \mathbb{K}$ (by adding random contributions from all players). Then we open $w - r$, compute its coordinates in public and add them to $[r_1], ..., [r_u]$ to get $[w_1], ..., [w_u]$. Finally linearity of the secret sharing implies we apply M to the coordinates by only local computation to get a secret sharing of the result $[\boldsymbol{x} \cdot \boldsymbol{y}]$. We can assume that each party P_j has also secret shared a bit b_j where $b_j = 1$ if and only if he is to get the result. We can then compute $[b_j s]$ for each j and open this privately to P_j.

Let us compute the cost of all this: the main part of the computation is to compute $[w]$ from sharings of the inputs. This requires essentially $\lceil In/k \rceil$ secure multiplications which the protocol from [DN07] can do at communication cost $\lceil In/k \rceil \cdot n$ elements in $\mathbb{K}$. An element in $\mathbb{K}$ has u bits and u is $O(k)$. So the cost in bits is $O(In/k \cdot n \cdot k) = O(In^2)$. One easily sees that the cost of sharing the inputs initially is also $O(In^2)$. The final stage where we go from $[w]$ to the result does not depend on I and its cost can therefore be ignored for all large enough I. □

For values of t that are smaller than the maximal value, the protocol in the above proof can be optimized in a straightforward way using packed secret sharing. Concretely, if $n = 2t + \ell$, one can secret share a vector of ℓ values where shares are only 1 field element, so this saves a factor ℓ compared to the original protocol. This way, we easily obtain an upper bound matching the result from Theorem 1.

5.2 Dishonest Majority

In this section, we sketch a generalization of known protocols in the preprocessing model leading to an upper bound that matches our lower bound for the $PIR_{I,n}$ function.

Let us consider a passively secure variant of the well known SPDZ protocol for $n = t + 1$, i.e., the secret values are additively secret shared among the players (no authentication is needed because we consider passive security). Linear operations can be done with no communication and multiplications are done using multiplication triples that are taken from the preprocessed data. It is clear that such a protocol would work with any linear secret sharing scheme as long as corruption of t players gives no information the secret.

So for the case of $n = t+s$, we can use Shamir secret-sharing with polynomials of degree t. Using the packed secret-sharing technique we can then encode a vector of $\Theta(s)$ values as the secret, instead of one value. This allows us to perform $\Theta(s)$ multiplications in parallel while communicating only $O(n)$ field elements. Namely, a multiplication involves opening two values, and this is done by sending shares to one player who reconstructs and sends the result to all parties.

Now, if we consider computing the $PIR_{I,n}$ function, the dominating part is to compute the storage access function (see Section 4). This function has a logarithmic depth layered circuit of size $O(In)$. We can therefore compute it by doing s operations in parallel at a time, leading to a communication complexity of $O(In^2/s)$ field elements.

One caveat is that this protocol will need a field with at least n elements, and the $PIR_{I,n}$ function is defined on binary values. This leads to an overhead factor of $\lg n$. However, using reverse multiplication friendly embeddings as in the previous subsection, we can get rid of this overhead.

Since we only need to consider $n > t \geq n/2$ in this model, we can assume that n is $\Theta(t)$, so a communication complexity of $O(In^2/s)$ bits matches our lower bound $\Omega(Int/s)$.

6 Conclusion and Future Work

In a nutshell, we have seen that nS where S is the input size, is a (up to a constant factor) lower bound on the communication complexity of unconditionally secure MPC, and for the particular functions we consider, this bound even equals the product of n and the circuit size of the function. For the dishonest majority case with preprocessing $O(nS)$ is also an upper bound (at least if one allows exponentially large storage for preprocessing).

Now, for honest majority, the obvious open problem is what happens for functions where the circuit size is much larger than the input: is there a lower bound that grows with the circuit size of the function (if we also require computational complexity polynomial in the circuit size)? Another question is whether our lower bound for suboptimal corruption threshold t is tight, in terms of the input size. Here $n = t + s$ and the bound is $\Omega(tS/s)$, so the question is if there is a matching upper bound, possibly by allowing exponential size preprocessing?

References

[BSPV99] Carlo Blundo, Alfredo De Santis, Giuseppe Persiano, and Ugo Vaccaro. Randomness complexity of private computation. *Computational Complexity*, 8(2):145–168, 1999.

[Can00] Ran Canetti. Security and composition of multiparty cryptographic protocols. *J. Cryptology*, 13(1):143–202, 2000.

[CCXY18] Ignacio Cascudo, Ronald Cramer, Chaoping Xing, and Chen Yuan. Amortized complexity of information-theoretically secure MPC revisited. In *CRYPTO (3)*, volume 10993 of *Lecture Notes in Computer Science*, pages 395–426. Springer, 2018.

[CK93] Benny Chor and Eyal Kushilevitz. A communication-privacy tradeoff for modular addition. *Inf. Process. Lett.*, 45(4):205–210, 1993.

[Cou18] Geoffroy Couteau. A note on the communication complexity of multiparty computation in the correlated randomness model. Cryptology ePrint Archive, Report 2018/465, 2018.

[DIK+08] Ivan Damgård, Yuval Ishai, Mikkel Krøigaard, Jesper Buus Nielsen, and Adam D. Smith. Scalable multiparty computation with nearly optimal work and resilience. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 241–261. Springer, 2008.

[DIK10] Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 445–465. Springer, 2010.

[DN07] Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2007.

[DNOR16] Ivan Damgård, Jesper Buus Nielsen, Rafail Ostrovsky, and Adi Rosén. Unconditionally secure computation with reduced interaction. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 420–447. Springer, 2016.

[DNPR16] Ivan Damgård, Jesper Buus Nielsen, Antigoni Polychroniadou, and Michael A. Raskin. On the communication required for unconditionally secure multiplication. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 459–488. Springer, 2016.

[DPP98] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Trans. Information Theory*, 44(3):1143–1151, 1998.

[DPP14] Deepesh Data, Manoj Prabhakaran, and Vinod M. Prabhakaran. On the communication complexity of secure computation. pages 199–216, 2014.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). pages 554–563, 1994.

[FY92] Matthew K. Franklin and Moti Yung. Communication complexity of secure computation (extended abstract). pages 699–710, 1992.

[GR03] Anna Gál and Adi Rosén. Lower bounds on the amount of randomness in private computation. pages 659–666, 2003.

[IKM+13] Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, Claudio Orlandi, and Anat Paskin-Cherniavsky. On the power of correlated randomness in secure computation. In *TCC*, volume 7785 of *Lecture Notes in Computer Science*, pages 600–620. Springer, 2013.

[KM97] Eyal Kushilevitz and Yishay Mansour. Randomness in private computations. *SIAM J. Discrete Math.*, 10(4):647–661, 1997.

[KR94] Eyal Kushilevitz and Adi Rosén. A randomnesss-rounds tradeoff in private computation. pages 397–410, 1994.

[Kus92] Eyal Kushilevitz. Privacy and communication complexity. *SIAM J. Discrete Math.*, 5(2):273–284, 1992.

[Weg87] Ingo Wegener. The complexity of boolean functions. https://eccc.weizmann.ac.il/static/books/The_Complexity_of_Boolean_Functions/, 1987.