Threshold Private Set Intersection with Better Communication Complexity

Satrajit Ghosh[1⋆] and Mark Simkin[2⋆⋆]

[1] Indian Institute of Technology Kharagpur
[2] Ethereum Foundation

Abstract. Given ℓ parties with sets $X_1, \ldots, X_\ell$ of size n, we would like to securely compute the intersection $\cap_{i=1}^{\ell} X_i$, if it is larger than $n-t$ for some threshold t, without revealing any other additional information. It has previously been shown (Ghosh and Simkin, Crypto 2019) that this function can be securely computed with a communication complexity that only depends on t and in particular does not depend on n. For small values of t, this results in protocols that have a communication complexity that is sublinear in the size of the inputs. Current protocols either rely on fully homomorphic encryption or have an at least quadratic dependency on the parameter t.
In this work, we construct protocols with a quasilinear dependency on t from simple assumptions like additively homomorphic encryption and oblivious transfer. All existing approaches, including ours, rely on protocols for computing a single bit, which indicates whether the intersection is larger than $n-t$ without actually computing it. Our key technical contribution, which may be of independent interest, takes any such protocol with secret shared outputs and communication complexity $\mathcal{O}(\lambda \ell \operatorname{poly}(t))$, where λ is the security parameter, and transforms it into a protocol with communication complexity $\mathcal{O}(\lambda^2 \ell t \operatorname{polylog}(t))$.

1 Introduction

In the private set intersection (PSI) setting, ℓ parties with private input sets $X_1, \ldots, X_\ell$ would like to jointly compute $\cap_{i=1}^{\ell} X_i$ without revealing anything else about any of the sets to each other. PSI is a powerful tool with applications in various places, such as botnet detection [NMH+10], online advertising [PSSZ15], private contact discovery [Mar14], and contact tracing [DPT20]. Various works have shown how to design asymptotically and practically efficient protocols in both the two and multiparty setting with security against both passive and active adversaries [Mea86, FNP04, KS05, DCW13, PSSZ15, KKRT16, PRTY19, PRTY20]. Unfortunately, all these protocols have communication complexities that are at least linear in the size of the smallest input set and it was observed by Freedman, Nissim, and Pinkas [FNP04] that one cannot hope to do better in general.

Ghosh and Simkin [GS19] have recently shown that the communication complexity can be sublinear in the sizes of the input sets, when the intersection is large. The authors considered the threshold private set intersection (TPSI) setting, where the parties would like to compute the intersection of their sets, if and only if it is larger than $n-t$, where n is the size of each set and t is some threshold. If the input sets do not have a large enough intersection, the protocols simply returns $\perp$ to the parties. Based on simple assumptions, such as the existence of oblivious transfer and additively homomorphic encryption, Ghosh and Simkin construct protocols for TPSI with a communication complexity of $\mathcal{O}(\lambda t^2 \operatorname{polylog} t)$ bits, where λ is the computational security parameter, for the two-party case. The authors also show how to construct a close to optimal two-party protocol based on fully homomorphic encryption with $\mathcal{O}(\lambda t \operatorname{polylog} t)$ bits and sketch how these protocols could be extended to the multiparty case. The authors show an $\Omega(t)$ lower bound on the communication complexity for the two-party case. Subsequently Branco, Döttling, and Pu [BDP21] present an ℓ-party protocol with a communication complexity of $\mathcal{O}(\lambda \ell t^2 \operatorname{polylog} t)$ bits based on threshold additively homomorphic encryption. Badrinarayanan et al. [BMRR21] propose a protocol for a setting similar to the

⋆ satrajit@cse.iitkgp.ac.in
⋆⋆ mark.simkin@ethereum.org

TPSI setting above, namely for computing the intersection of ℓ sets with a communication complexity of $\mathcal{O}(\lambda \ell t \operatorname{polylog} t)$, when $\left|(\cup_{i=1}^{\ell} X_i) \setminus (\cap_{i=1}^{\ell} X_i\right| \leq t$. For $\ell = 2$, the work of Badrinarayanan et al. is equivalent to two-party TPSI, but for $\ell > 2$ their work requires the set intersection to not only be large, but additionally they require that the parties have less than t distinct elements outside the intersection among all sets. Both Branco, Döttling, and Pu as well as Badrinarayanan et al. show that one cannot do better than $\Omega(\ell t)$ in their respective settings and provide, up to polylog factors, matching upper bounds based on fully homomorphic encryption.

All three works [GS19, BDP21, BMRR21] leave constructing asymptotically optimal multiparty protocols from other assumptions than the existence of fully homomorphic encryption as an open problem.

1.1 Applications of Threshold Private Set Intersection

As has been pointed out by the previous works [GS19, BDP21, BMRR21], threshold private set intersection is not just an interesting theoretical object to study, but also has the potential to be useful in a variety of practical applications, where parties are only interested in the actual intersection if it is indeed large. In the biometric authentication setting, we have a biometric reading represented as a feature vector and a template. An authentication attempt can directly be discarded, if the reading has a small intersection with the template. In the setting of ride sharing or dating apps, users may not care to share their private data with each other, if they do not have a large intersection.

Even protocols for general private set intersection can benefit from more efficient TPSI protocols. Parties that would like to compute the intersection of their sets can just execute a TPSI protocol with thresholds $2, 2^2, 2^4, \ldots$ until an execution returns the intersection instead of $\perp$. If the intersection of the input sets is large, then this PSI protocol has a communication complexity that is sublinear in the size of the input sets. This is in stark contrast to the majority of existing works on PSI that usually have a communication complexity that is at least linear in the smallest set size.

1.2 Our Contribution

In this work, we present new protocols for computing the threshold private set intersection among ℓ parties with a quasilinear rather than quadratic dependency on t from simple assumptions. More concretely, we construct protocols with a communication complexity of $\mathcal{O}(\lambda^2 \ell t \operatorname{polylog} t)$ bits. We follow the blueprint of Ghosh and Simkin [GS19] and tackle the problem by splitting it into two smaller problems. We first execute a private intersection cardinality test (PICT) protocol $\Pi^{n,t}_{\ell\text{-pict}}(X_1, \ldots, X_\ell)$ that checks, whether the given sets $X_1, \ldots, X_\ell$ have an intersection of size at least $n - t$. If they do, we can execute another protocol for computing the actual intersection in a communication efficient manner.

Computing the intersection, when it has already been established that it is indeed large enough, can be done generically from assumptions, like the existence of oblivious transfer or additively homomorphic encryption, with a close to optimal communication complexity of $\tilde{\mathcal{O}}(\lambda \ell t)$ bits as has been shown by Ghosh and Simkin. Thus, the main challenge and the focus of this work is to construct communication efficient PICT protocols from simple assumptions, which output a single bit that indicates, whether the intersection is large enough.

Our main technical contribution is a transformation that takes any PICT protocol with secret shared outputs[3] and communication complexity $\mathcal{O}(\lambda \ell \operatorname{poly}(t))$ and transforms it into a new protocol that solves the same task, but has a communication complexity of only $\mathcal{O}(\lambda^2 \ell t \operatorname{polylog}(t))$. An implication of this compiler is the existence of multiparty protocols with the above stated communication complexity from effectively any assumption that implies secure computation. The efficiency of a protocol that is given as input to our transformation only affects the constant in the $\operatorname{polylog}(t)$ exponent.

[3] All existing protocols can easily be adapted to output secret shares of the output instead of the output itself.

Is This Stuff Practical? We stress that the main focus of this work is to construct asymptotically more efficient protocols from simple assumptions. In particular, the goal is achieve a communication complexity that has a quasi-linear and not a quadratic dependency on the threshold t. We hope that our work will eventually lead to practically efficient protocols, but we think that our current results are still too inefficient for most reasonable real-world parameters. We leave constructing concretely efficient protocols as an exciting open problem for future work. Nonetheless, we view our work as a significant theoretical step towards more efficient protocols for threshold private set intersection.

1.3 Technical Overview

For the sake of this overview, let us focus on the two-party case. We would like to design a protocol that takes two sets $X, Y \subset U$ from some universe U as input and outputs a bit that indicates, whether $|X \cap Y| \geq n - t$ or equivalently, whether the symmetric set difference $|X \triangle Y| := |X \setminus Y \cup Y \setminus X| \leq 2t$. Our main idea is to approach this problem via a divide and conquer strategy, i.e. to partition the sets X and Y into smaller sets $X_1, \ldots, X_t$ and $Y_1, \ldots, Y_t$ and then to perform a series of independent PICTs on each pair X_i and Y_i for $i \in [t] := \{1, \ldots, t\}$.

More precisely, imagine we have random functions[4] $H^i : U \to [t]$ for $i \in [\epsilon]$ for some value ϵ that take set elements as input and outputs values in $[t]$. Define $X_i^j = \{x \mid x \in X \wedge H^j(x) = i\}$ and $Y_i^j = \{y \mid y \in Y \wedge H^j(y) = i\}$ for $i \in [t]$ and $j \in [\epsilon]$ and observe that for all $j \in [\epsilon]$

$$|X \triangle Y| = \sum_{i=1}^{t} \left| X_i^j \triangle Y_i^j \right|.$$

Consider some fixed $j \in [\epsilon]$. If $|X \triangle Y| \leq 2t$, then in expectation each pair of sets X_i^j and Y_i^j contains at most two elements in their symmetric set difference and one can show that (for a fixed j) with a constant probability none of the pairs has a symmetric set difference that is larger than $\mathcal{O}(\ln t)$. It follows that when $|X \triangle Y| \leq 2t$, there must exist at least one j for which $\left| X_i^j \triangle Y_i^j \right| \in \mathcal{O}(\ln t)$ for all $i \in [t]$ with an overwhelming in ϵ probability.

So how is this helpful? Imagine we were given access to an auxiliary functionality $\mathcal{F}_{\triangle}^{n,\tilde{t},v}$ that takes two sets as input and either returns a secret sharing of the size of their *exact* symmetric set difference or a secret sharing of some default value v, if the symmetric set difference is larger than $\tilde{t} \approx \ln t$. We can use $\mathcal{F}_{\triangle}^{n,\tilde{t},v}$ on each of the ϵt many subset pairs to obtain equally many secret shared values and then add all the values together that belong to inputs, which were partitioned using the same random partitioning function to get a total of ϵ many secret shared sums. Each of those sums either equals the exact size of the symmetric set difference of X and Y or some value, which has v as a summand. By picking $v = t + 1$, we ensure that each sum containing v is larger than t. As the final step in our protocol, we run a generic secure computation protocol for checking, whether any of the ϵ sums is at most t in which case we conclude that the inputs X and Y have a large enough intersection.

To make our protocol work, we still need to instantiate $\mathcal{F}_{\triangle}^{n,\tilde{t},v}$. We show that this can be done from any PICT protocol with secret shared outputs for thresholds $\tilde{t}$. If the given protocol has a communication complexity of $\mathcal{O}(\lambda \operatorname{poly}(\tilde{t}))$ bits, then our instantiation of $\mathcal{F}_{\triangle}^{n,\tilde{t},v}$ has a communication complexity of $\mathcal{O}(\lambda \tilde{t} \operatorname{poly}(\tilde{t})) = \mathcal{O}(\lambda \ln t \operatorname{polylog} t) = \mathcal{O}(\lambda \operatorname{polylog} t)$. Since our approach only relies on generic secure computation and existing PICT protocols, it follows that we can instantiate our constructions from assumptions that imply both of these cryptographic objects. As we will see, this means that we can instantiate our results from oblivious transfer or generic additively homomorphic encryption.

Our multiparty PICT protocols follows the same blueprint as the protocol outlined above, but need to overcome several other challenges. In the the two-party case we got away with just talking about the

[4] Throughout the paper we will use random functions for the sake of simplicity, but we stress that all of our constructions and arguments work equally well with pseudorandom functions, where the key is known to all parties.

symmetric set difference, since an upper bound on the difference directly translates into a lower bound on the set intersection size. In the multiparty setting this is not the case any longer and we will need to directly talk about the set intersection sizes in all the buckets instead. While it may sounds like a minor change, it does introduce quite some small technical challenges that we will highlight in more detail and then overcome in Section 4.

Paper Outline. In Section 2 we recall some basic preliminaries and define all the required notation that will be needed throughout the paper. In Section 3, we present our protocol for the two-party case. We stress that this *does not* asymptotically improve upon the state-of-the-art, which has a communication complexity of $\mathcal{O}(\lambda t \operatorname{polylog} t)$ bits[5]. We do, however, believe that our two-party protocol highlights the main ideas of this work quite well, while avoiding some of the complexities that come from considering multiple parties. In Section 4 we present our multiparty protocol, which is the main technical contribution of this work.

2 Preliminaries

Notation. We write $[n] = \{1, 2, \dots, n\}$. Let $\log x$ be the logarithm of x with base 2 and $\ln x$ the one with base e. For convenience, we assume the natural numbers start at one, i.e. $\mathbb{N} = \{1, 2, 3, \dots\}$. Let λ be the computational and ϵ the statistical security parameter and we assume that $\epsilon/\lambda \in \mathcal{O}(1)$. We write $\mathbb{F}$ to denote a finite field of prime order and we assume that $|\mathbb{F}| \geq 2^\epsilon$. For parties $P_1, \dots, P_\ell$ with inputs $X_1, \dots, X_\ell$ that have oracle access to an ideal functionality $\mathcal{F}$, we write $(b_1, \dots, b_\ell) \leftarrow \mathcal{F}(X_1, \dots, X_\ell)$ as a shorthand notation for each party i sending X_i to the ideal functionality and, once all inputs are received, receiving back output b_i. For a protocol Π, we write $\mathtt{CC}(\Pi)$ to denote the communication complexity of Π, i.e. the number of bits exchanged in one execution of the protocol.[6]

Theorem 1 (Chernoff Bound). *Let $I_1, \dots, I_n$ be random variables with $0 \leq I_i \leq 1$ for all $i \in [n]$. Define $I = \sum_{i=1}^{n} I_i$ and let $\mu = \mathbb{E}[I]$. For any $\delta \geq 1$,*

$$\Pr[I \geq (1+\delta)\mu] \leq e^{-\frac{\delta\mu}{3}}.$$

Set Gymnastics. Let U be the universe from which set elements will be sampled and let $Z = (z_1, z_2, \dots)$ be an auxiliary (sorted) universe such that $U \cap Z = \emptyset$. We will use upper case letter for sets and lower case letters for their elements, e.g. $S = \{s_1, \dots, s_n\}$. For $S \in U^n$ and function $H : U \to [t]$, we write $(S_1, \dots, S_t) \leftarrow H(S)$ as a shorthand notation to specify the sets $S_i = \{s \mid s \in S \wedge H(s) = i\}$.

Secret Sharing. Let $\mathtt{Share}_n : \mathbb{F} \to \mathbb{F}^\ell$ be an ℓ-out-of-ℓ secret sharing algorithm that takes $v \in \mathbb{F}$ as input and outputs uniformly random $v_1, \dots, v_\ell \in \mathbb{F}$, such that $v = \sum_{i=1}^{\ell} v_i$. When an algorithm or a functionality outputs $\mathtt{Share}_\ell(v)$, we mean that party i receives shares v_i.

2.1 Secure Multiparty Computation

We assume familiarity with standard secure computation notions in the standalone model (see [Lin17]). In this paper, we assume that all parties are pairwise connected via a synchronous network and authenticated private channels. Additionally the parties have access to a broadcast channel and we assume that sending a message on this channel has a unit cost. We consider a static adversary that can corrupt all but one parties passively.

[5] This communication complexity can be obtained, without using fully homomorphic encryption, by using the construction of Ghosh and Simkin [GS19] in combination with an observation due to Badrinarayanan et al. [BMRR21].

[6] We assume that the communication complexity is a deterministic function of the inputs and parameters of Π.

2.2 Private Intersection Cardinality Testing.

For the two-party case the functionality we are interested in is the $\mathcal{F}_{\mathsf{pict}}^{n,t}(X,Y)$ functionality shown in Figure 1. It is helpful to note that for X and Y with $n = |X| = |Y|$, it holds that

$$|X \cap Y| \leq n - t \iff |X \triangle Y| > 2t,$$

which means that the functionality outputs a sharing of 1 for two sets of size n if and only if $|X \cap Y| > n - t$. The functionality $\mathcal{F}_{\mathsf{pict}}^{n,t}(X,Y)$ does allow for the input sets to be of unequal sizes smaller than n in which case the equivalence above does not hold. This is done for the sake of simplifying the presentation of our construction in the two-party case. The multiparty functionality will be introduced in Section 4 and it will require the input sets to be of the same size.

Functionality $\mathcal{F}_{\mathsf{pict}}^{n,t}(X,Y)$

if $|X| > n$ **or** $|Y| > n$
  return $\perp$
if $|X \triangle Y| > t$
  return $\mathtt{Share}_2(0)$
else
  return $\mathtt{Share}_2(1)$

Fig. 1. Functionality takes two sets X and Y of size at most n as input and checks whether $|X \triangle Y| \leq t$.

2.3 Some Auxiliary Functionalities

In the following, we define some helpful functionalities that will come in handy later on. They can be realized using any generic secure computation protocol and will not affect our communication complexities in any meaningful way.

The functionalities in Figure 2 allow for comparing a secret shared input against a publicly known threshold and returning either the secret shared value or a default value. Both functionalities can be easily realized with communication complexities that are linear in their input length with standard secure computation tools.

Functionality $\mathcal{F}_{\mathsf{cmp}}^{t,v}(r_1, r_2)$

if $r_1 + r_2 = t$
  return $\mathtt{Share}_2(v)$
else
  return $\mathtt{Share}_2(r_1 + r_2)$

Functionality $\mathcal{F}_{\ell\text{-}\mathsf{geq}}^{t,v}(r_1, \ldots, r_\ell)$

Compute $s := \sum_{i=1}^{\ell} r_i$
if $s \geq t$
  return $\mathtt{Share}_\ell(s)$
else
  return $\mathtt{Share}_\ell(v)$

Fig. 2. Some useful private comparison function of secret shared inputs.

Functionality $\mathcal{F}^{t,\epsilon}_{\ell\text{-vec-leq}}((s^1_1, \dots s^\epsilon_1), \dots, (s^1_\ell, \dots s^\epsilon_\ell))$ in Figure 3 takes ϵ many ℓ-out-of-ℓ secret shared field elements as input and returns 1 if any one of them is smaller than t and 0 otherwise. This functionality can be realized using generic secure computation with a communication complexity of $\mathcal{O}(\epsilon\ell|\mathbb{F}|)$ bits.

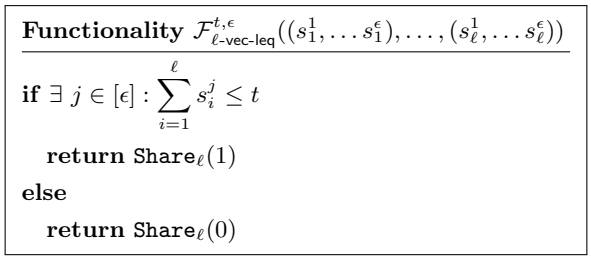

Fig. 3. Functionality for checking, whether one of the ϵ many secret shared inputs is at most t.

The functionality in Figure 4 computes the minimum among a list of input values and returns that value in secret shared form.

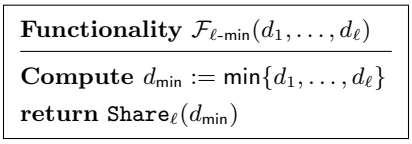

Fig. 4. Functionality for computing a secret sharing of the minimum among a set of inputs.

The functionality in Figure 5 checks whether at least one of multiple secret shared values is within a given interval.

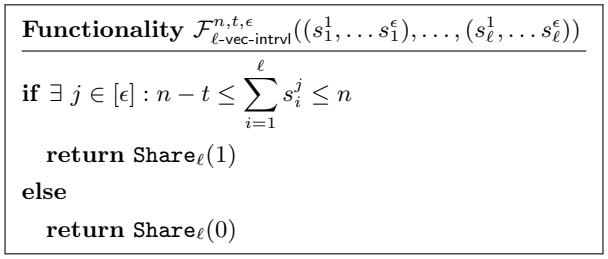

Fig. 5. Functionality for checking, whether at least one of ϵ many secret shared input values is in between $n - t$ and n.

3 The Two-Party Divide-and-Conquer Approach

In this section, we will focus on the two-party case for the sake of presenting our main ideas in a simplified setting.

Let us begin with a simple lemma, which states that one can partition sets X and Y into smaller sets and compute the size of their symmetric set difference in a divide-and-conquer fashion.

Lemma 1. *Let $X, Y \subset U$, let $t \in \mathbb{N}$, and let $H : U \to [t]$ be an arbitrary function. For $(X_1, \ldots, X_t) \leftarrow H(X)$ and $(Y_1, \ldots, Y_t) \leftarrow H(Y)$, it holds that*

$$|X \triangle Y| = \sum_{i=1}^{t} |X_i \triangle Y_i|.$$

Proof. Consider two arbitrary sets $X, Y \subset U$. We observe that $|X \triangle Y| = |X \setminus Y| + |Y \setminus X|$. If $v \in X \setminus Y$, then there exists an index $i \in [t]$ such that $v \in X_i \setminus Y_i$ and since $X_i \cap X_j = \emptyset$ for any $j \neq i$, it holds that i is unique. The other way round, for any $i \in [t]$ and any $v \in X_i \setminus Y_i$, it holds that $v \in X \setminus Y$. Thus

$$|X \setminus Y| = \sum_{i=1}^{t} |X_i \setminus Y_i|$$

and by symmetry of the above argument

$$\begin{aligned} |X \triangle Y| =& |X \setminus Y| + |Y \setminus X| \\ =& \sum_{i=1}^{t} |X_i \setminus Y_i| + \sum_{i=1}^{t} |Y_i \setminus X_i| \\ =& \sum_{i=1}^{t} |X_i \setminus Y_i| + |Y_i \setminus X_i| = \sum_{i=1}^{t} |X_i \triangle Y_i|. \end{aligned}$$

□

Next, we observe that, if the symmetric set difference of X and Y is at most t, then the symmetric set difference of each pair of subsets X_i and Y_i for $i \in [t]$ is in $\mathcal{O}(\ln t)$ with a constant probability.

Lemma 2. *Let $n, t, \tilde{t} \in \mathbb{N}$ with $t < n$ and $\tilde{t} \geq 1 + 3 \ln 2t$. Let $H : U \to [t]$ be a random function, and let $X, Y \in U^n$. If $|X \triangle Y| \leq t$, then for $(X_1, \ldots, X_t) \leftarrow H(X)$ and $(Y_1, \ldots, Y_t) \leftarrow H(Y)$ it holds that*

$$\Pr\left[\exists i \in [t] : |X_i \triangle Y_i| \geq \tilde{t}\right] \leq 1/2,$$

where the probability is taken over the random choice of H.

Proof. Assume that $|X \triangle Y| \leq t$. For all $i \in [t]$, it holds that $X_i \triangle Y_i = \{v \mid v \in X \triangle Y \wedge H(v) = i\} \subset X \triangle Y$. Fix one bucket j and let I_v be the indicator variable for whether $v \in X \triangle Y$ landed in bucket j or not. For

$$\mathbb{E}[|X_j \triangle Y_j|] = \mathbb{E}\left[\sum_{v \in X \triangle Y} I_v\right] = \sum_{v \in X \triangle Y} \mathbb{E}\left[I_v\right] = \sum_{v \in X \triangle Y} 1/t \leq 1$$

we get by Chernoff bound that

$$\Pr\left[|X_j \triangle Y_j| \geq 1 + 3 \ln 2t\right] \leq e^{-\frac{3 \ln 2t}{3}} = 1/2t,$$

where the probability is taken over the random choice of the function H. The statement follows by union bounding over all t buckets. □

Now, if $|X \triangle Y| \leq t$ and we partition sets X and Y not once, but ϵ many times, then we are guaranteed with overwhelming probability that at least one of those partitions has no bucket that contains more than $\mathcal{O}(\ln t)$ elements.

Theorem 2. *Let $n, t, \tilde{t} \in \mathbb{N}$ with $t < n$ and $\tilde{t} \geq 1 + 3\ln 2t$. For each $i \in [\epsilon]$, let $H^i : U \to [t]$ be a random function. Let $X, Y \in U^n$ be two sets of size n and $(X_1^i, \ldots, X_t^i) \leftarrow H^i(X)$ and $(Y_1^i, \ldots, Y_t^i) \leftarrow H^i(Y)$ for $i \in [\epsilon]$. If $|X \triangle Y| \leq t$, then*

$$\Pr\left[\exists i_1, \ldots i_\epsilon \in [t] : \left|X_{i_j}^j \triangle Y_{i_j}^j\right| \geq \tilde{t} \ \forall j \in [\epsilon]\right] \leq 2^{-\epsilon},$$

where the probability is taken over the random choice of $H^1, \ldots, H^\epsilon$.

Proof. Assume $|X \triangle Y| \leq t$, then

$$\begin{aligned}
&\Pr\left[\exists i_1, \ldots i_\epsilon \in [t] : \left|X_{i_j}^j \triangle Y_{i_j}^j\right| \geq 1 + 3\ln 2t \ \forall j \in [\epsilon]\right] \\
=&\prod_{j=1}^{\epsilon} \Pr\left[\exists i_j \in [t] : \left|X_{i_j}^j \triangle Y_{i_j}^j\right| \geq 1 + 3\ln 2t\right] \\
\leq&\prod_{j=1}^{\epsilon} 1/2 = 2^{-\epsilon},
\end{aligned}$$

where the last inequality follows from Lemma 2.

□

From the above it now follows that, if there exists at least a single bucket in each of the ϵ partitions, which contains more than $1 + 3\ln 2t$ elements of the symmetric set difference, then we can conclude that $|X \triangle Y| > t$ with overwhelming probability.

Corollary 1. *Let $n, t, \tilde{t} \in \mathbb{N}$ with $t < n$ and $\tilde{t} \geq 1 + 3\ln 2t$. For each $i \in [\epsilon]$, let $H^i : U \to [t]$ be a random function. Let $X, Y \in U^n$ be two sets of size n and $(X_1^i, \ldots, X_t^i) \leftarrow H^i(X)$ and $(Y_1^i, \ldots, Y_t^i) \leftarrow H^i(Y)$ for $i \in [\epsilon]$. If there exist indices $i_1, \ldots i_\epsilon \in [t]$, such that for all $j \in [\epsilon]$ it holds that $\left|X_{i_j}^j \triangle Y_{i_j}^j\right| \geq \tilde{t}$, then*

$$\Pr[|X \triangle Y| > t] \geq 1 - 2^{-\epsilon},$$

where the probability is taken over the random choices of $H^1, \ldots H^\epsilon$.

Functionality $\mathcal{F}_{\triangle}^{n,t,v}(X, Y)$

if $|X| > n$ **or** $|Y| > n$
  return $\perp$
if $|X \triangle Y| > t$
  return $\texttt{Share}_2(v)$
else
  return $\texttt{Share}_2(|X \triangle Y|)$

Fig. 6. Functionality for computing the exact symmetric set difference, if it is smaller than t, of sets X and Y with elements from U. The sets X and Y may be of different sizes, but neither of them is larger than n.

Armed with the above observations, we are now ready to present our construction. The description of our protocol makes use of an ideal functionality $\mathcal{F}_{\triangle}^{n,t,v}$ (see Figure 6) that takes two sets as input and either returns a secret sharing of their symmetric set difference or returns a sharing of some value v. The sets may be of different sizes, but are both not larger than n. We want to highlight that allowing for input sets of

unequal size is only possible, because we are currently talking about the symmetric set difference. Looking ahead, we will be directly talking about the size of the intersection in the multiparty protocols in Section 4 and therefore we will need to take care of making the sets be of the correct and same size. We show how to instantiate $\mathcal{F}_{\triangle}^{n,t,v}$ in Section 3.1

Construction $\Pi_{\mathsf{pict}}^{n,t}(X,Y)$

for $i \in [\epsilon]$:
  Alice: $(X_1^i, \ldots, X_t^i) \leftarrow H^i(X)$
  Bob: $(Y_1^i, \ldots, Y_t^i) \leftarrow H^i(Y)$
  for $j \in [t]$:
    $(r_j^i, s_j^i) \leftarrow \mathcal{F}_{\triangle}^{n,\tilde{t},t+1}\left(X_j^i, Y_j^i\right)$
  Alice: $r^i := \sum_{j=1}^{t} r_j^i$
  Bob: $s^i := \sum_{j=1}^{t} s_j^i$
$(r,s) \leftarrow \mathcal{F}_{\mathsf{2\text{-}vec\text{-}leq}}^{t,\epsilon}\left((r^1, \ldots, r^\epsilon), (s^1, \ldots, s^\epsilon)\right)$
return (r,s)

Fig. 7. Protocol for private intersection cardinality testing.

Theorem 3. *Let $n, t, \tilde{t} \in \mathbb{N}$ with $n > t$ and $\tilde{t} = 1 + 3\ln 2t$. The protocol Π_{pict} depicted in Figure 7 securely realizes $\mathcal{F}_{\mathsf{pict}}^{n,t}$ using $\epsilon \cdot t$ calls to $\mathcal{F}_{\triangle}^{n,\tilde{t},t+1}$ and one call to $\mathcal{F}_{\mathsf{vec\text{-}cmp}}^{t,\epsilon}$.*

Proof. We prove correctness and privacy separately.

Correctness. Let $X_1, \ldots, X_n$ be arbitrary sets. If $|X \triangle Y| > t$, then the protocol always produces the correct output. Either there exist $j_1, \ldots, j_\epsilon \in [t]$, such that $\left|X_{j_i}^i \triangle Y_{j_i}^i\right| > \tilde{t}$ for all $j \in [\epsilon]$, which means that $r_{j_i}^i + s_{j_i}^i = t+1$ and thus also $r^i + s^i > t$ for all $i \in [\epsilon]$ or the protocol computes the size of the symmetric set difference correctly in which case $\mathcal{F}_{\mathsf{2\text{-}vec\text{-}leq}}^{t,\epsilon}\left((r^1, \ldots, r^\epsilon), (s^1, \ldots, s^\epsilon)\right)$ produces the correct output.

If $|X \triangle Y| \leq t$, then the protocol can produce an incorrect output, if there exists indices $j_1, \ldots, j_\epsilon \in [t]$ such that $\left|X_{j_i}^i \triangle Y_{j_i}^i\right| > \tilde{t}$ for all $j \in [\epsilon]$. By Theorem 2, we know that this happens with probability at most $2^{-\epsilon}$ and thus the protocol produces the correct output with probability at least $1 - 2^{-\epsilon}$.

Privacy. Without loss of generality assume that Alice is corrupted. At each step of the protocol, she only sees one share of freshly independent secret shared values returned by the ideal functionalities. Her view can simply be simulated by providing her shares of independent secret sharings of 0 instead of the real values. The indistinguishability of Alice's simulated view trivially follows from the indistinguishability of the secret sharing scheme.

□

3.1 Instantiating $\mathcal{F}_{\triangle}^{n,t,v}$

To instantiate $\mathcal{F}_{\triangle}^{n,t,v}$, we simply use $\mathcal{F}_{\mathsf{pict}}^{n,i}$ once for each threshold $i \in [t]$ and then accumulate the result.

Construction $\Pi_{\triangle}^{n,t,v}(X,Y)$

1 : **for** $i \in [t]$:
2 : $\quad (r_i, s_i) \leftarrow \mathcal{F}_{\mathsf{pict}}^{n,i}(X,Y)$
3 : **Alice:** $r := t + 1 - \sum_{j=1}^{t} r_i$
4 : **Bob:** $s := -\sum_{j=1}^{t} s_j$
5 : $(d_1, d_2) \leftarrow \mathcal{F}_{\mathsf{cmp}}^{t,v}(r,s)$
6 : **return** (d_1, d_2)

Fig. 8. Protocol $\Pi_{\triangle}^{n,t,v}$ realizing $\mathcal{F}_{\triangle}^{n,t,v}$.

Theorem 4. *Let $n, t \in \mathbb{N}$ with $n > t$ and $v \in \mathbb{F}$. The protocol $\Pi_{\triangle}^{n,t,v}$ depicted in Figure 8 securely implements $\mathcal{F}_{\triangle}^{n,t,v}$ using one call to $\mathcal{F}_{\mathsf{pict}}^{n,i}$ for each $i \in [t]$.*

Proof. For correctness, we observe that $\mathcal{F}_{\mathsf{pict}}^{n,i}(X,Y)$ outputs a sharing of 1, when $|X \triangle Y| \leq i$. Thus, if $|X \triangle Y| \leq t$, we have that (r,s) is a secret sharing of exactly $|X \triangle Y|$ and if $|X \triangle Y| > t$, then (r,s) is a secret sharing of t.

Seeing that the protocol is secure is straightforward, assume that Alice is corrupted. To simulated the responses of $\mathcal{F}_{\mathsf{pict}}^{n,i}(X,Y)$, we add shares of a secret sharing of 0 to her view. Given the output of the functionality, we secret share that value and add one share to Alice's view. It is straightforward to see that this perfectly simulates Alice's view in the real world, which completes the proof.

□

To instantiate our overall protocol, we now need to instantiate the $\mathcal{F}_{\mathsf{pict}}^{n,t}$ functionality that is being used inside of $\Pi_{\triangle}^{n,t,v}$. The original two-party PICT protocols of Ghosh and Simkin [GS19] require the input sets to be of the same size and the output is not secret shared. Their protocols, however, work equally well for sets of different sizes, can easily return secret shared output values, and thus can be used to instantiate our functionality $\mathcal{F}_{\mathsf{pict}}^{n,t}$. Internally, their work relies on a protocol for securely computing the determinant of a secret shared matrix. They instantiate that protocol with a communication complexity of $\mathcal{O}(\lambda t^2 \operatorname{polylog}(t))$ via additively homomorphic encryption, but using a protocol for computing that determinant by Cramer and Damgård [CD01], one can instantiate the protocol of Ghosh and Simkin with communication complexity $\mathcal{O}(\lambda \ln^3 t)$ from generic secure computation. It follows that our result can be instantiated from any assumption, such as the existence of additively homomorphic encryption or oblivious transfer, that implies secure computation.

In our instantiation, we have ϵt buckets and for each of them we execute the protocol of Ghosh and Simkin $\mathcal{O}(\ln t)$ times with a threshold of $\mathcal{O}(\ln t)$. Thus we get the following corollaries.

Corollary 2. *Assuming the existence of oblivious transfer (or additively homomorphic encryption), there exists a constant-round protocol for securely computing the two-party private intersection cardinality test for threshold t with communication complexity of $\mathcal{O}(\epsilon^2 t \operatorname{polylog} t)$ bits.*

Combining the results in our paper with the protocols for actually computing the intersection, once it is known that it is large enough, from by Ghosh and Simkin [GS19], we get the following result.

Corollary 3. *Assuming the existence of oblivious transfer (or additively homomorphic encryption), there exists a constant-round protocol for threshold private set intersection among two parties with threshold t with communication complexity of $\mathcal{O}(\epsilon^2 t \operatorname{polylog} t)$ bits.*

4 The Multiparty Case

We now proceed to present our protocol for the multiparty case, which follows the blueprint from Section 3, but needs to overcome several additional challenges. The functionality that we would like to realize in this section is depicted in Figure 9.

Functionality $\mathcal{F}^{n,t}_{\ell\text{-pict}}(X_1, \ldots, X_\ell)$

if $|X_1| \neq n$ **or** $\ldots$ **or** $|X_\ell| \neq n$
  return $\perp$
if $\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n - t$
  return $\texttt{Share}_\ell(1)$
else
  return $\texttt{Share}_\ell(0)$

Fig. 9. Functionality for multiparty private intersection cardinality testing among sets of *size exactly* n.

In the two-party case we got away with talking about the set difference as a surrogate for the size of the set intersection due to the equivalence of intersection size and size of the symmetric set difference that is pointed out in Section 2.2. In the multiparty case, we make no such assumption.

To call a protocol for computing the size of the intersection in each bucket among ℓ parties, we will now ensure that the sets in each bucket are of the same size. We achieve this by padding sets with elements from an (ordered) auxiliary universe $Z = \{z_1, z_2, \ldots\}$ with $Z \cap U = \emptyset$. For $n, b \in \mathbb{N}$ with $b > n$ and any set $X \in U^n$, we define $\texttt{Pad}(X, b) := X \cup \{z_i \mid i \in [n - b]\}$. Lemma 3 shows the relationship between the size of the intersection among padded sets and unpadded sets.

Lemma 3. *Let $b \in \mathbb{N}$ and let $X_1, \ldots, X_\ell \subset U$ with $|X_i| \leq b$ for all $i \in [\ell]$. Let $d_i := ||X_i| - b|$ for $i \in [\ell]$. Then*

$$\left|\bigcap_{i=1}^{\ell} X_i\right| = \left|\bigcap_{i=1}^{\ell} \texttt{Pad}(X_i, b)\right| - \min(d_1, \ldots, d_\ell)$$

Proof. Define $W_i = \texttt{Pad}(X_i, b) \setminus X_i$ for $i \in [\ell]$. We observe that

$$\left|\bigcap_{i=1}^{\ell} \texttt{Pad}(X_i, b)\right| = \left|\bigcap_{i=1}^{\ell} (X_i \cup W_i)\right| = \left|\bigcap_{i=1}^{\ell} X_i\right| + \left|\bigcap_{i=1}^{\ell} W_i\right| = \left|\bigcap_{i=1}^{\ell} X_i\right| + \min(d_1, \ldots, d_\ell),$$

where the second equality follows from the fact that $Z \cap U = \emptyset$ and thus the lemma statement follows. □

The following Lemma can be seen as a generalization of Lemma 1 to the multiparty case. On an intuitive level, it states that a lower bound on the size of the intersection of ℓ sets translates into a lower bound on the cumulative size of the intersections in each buckets

Lemma 4. *Let $n, \ell, t \in \mathbb{N}$ with $t < n$, let $H : U \to [t]$ be a random function, let $X_1, \ldots, X_\ell \in U^n$, and $(X_{i,1}, \ldots, X_{i,t}) \leftarrow H(X_i)$ for $i \in [\ell]$. It holds that*

$$\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n - t$$

if and only if

$$\sum_{k=1}^{t}\left|X_{j,k} \setminus \bigcap_{i=1}^{\ell} X_{i,k}\right| \leq t, \forall j \in [\ell].$$

Proof. Let $W_{j,k} := X_{j,k} \setminus \bigcap_{i=1}^{\ell} X_{i,k}$ for $k \in [t]$ and $j \in [\ell]$. We observe that for each pair $k, k' \in [t]$, it holds that $W_{j,k} \cap W_{j,k'} = \emptyset$ and thus

$$\left|X_j \setminus \bigcap_{i=1}^{\ell} X_i\right| = \sum_{k=1}^{t}\left|X_{j,k} \setminus \bigcap_{i=1}^{\ell} X_{i,k}\right|.$$

The statement follows from the fact that

$$\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n - t,$$

if and only if

$$\left|X_j \setminus \bigcap_{i=1}^{\ell} X_i\right| \leq t, \forall j \in [\ell].$$

□

Similarly to Theorem 2, we will now show that, if the intersection is large enough, then there exists an index $j \in [\epsilon]$ such that the partitioning with H^j will not result in any one party having too many elements in a single bucket that do not belong to that buckets intersection.

Theorem 5. *Let $n, \ell, t, \tilde{t} \in \mathbb{N}$ with $t < n$ and $\tilde{t} \geq 1 + 3\ln(2t\ell)$. For each $j \in [\epsilon]$, let $H^j : U \to [t]$ be a random function. Let $X_1, \ldots, X_\ell \in U^n$ be sets of size n and $(X^j_{i,1}, \ldots, X^j_{i,t}) \leftarrow H^j(X_i)$ for $j \in [\epsilon]$ and $i \in [\ell]$. If*

$$\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n - t,$$

then

$$\Pr\left[\exists \begin{array}{c} k_1, \ldots, k_\epsilon \in [t] \\ i_1, \ldots, i_\epsilon \in [\ell] \end{array} : \left|X^j_{i_j,k_j} \setminus \bigcap_{m=1}^{\ell} X^j_{m,k_j}\right| \geq \tilde{t}\ \forall j \in [\epsilon]\right] \leq 2^{-\epsilon},$$

where the probability is taken over the random choice of $H^1, \ldots, H^\epsilon$.

Proof. Let $W_i := X_i \setminus \left(\bigcap_{m=1}^{\ell} X_m\right)$ for $i \in [\ell]$. Assume $\left|\bigcap_{m=1}^{\ell} X_m\right| > n - t$, then for each $i \in [\ell]$, it holds that $|W_i| \leq t$. Fix some $i \in [\ell]$, $j \in [\epsilon]$, $k \in [t]$ and consider $X^j_{i,k}$, where $(X^j_{i,1}, \ldots, X^j_{i,t}) \leftarrow H^j(X_i)$. For $v \in W_i$, let I_v be the indicator variable for whether $v \in X^j_{i,k}$ or not. Then,

$$\mathbb{E}\left[\sum_{v \in W_i} I_v\right] = \sum_{v \in W_i} 1/t \leq 1$$

and thus by Chernoff bound

$$\Pr\left[\sum_{v \in W_i} I_v \geq 1 + 3\ln(2t\ell)\right] \leq e^{-\frac{3\ln(2t\ell)}{3}} = 1/2t\ell.$$

By union bound over all t buckets and all ℓ sets, we can thus conclude that

$$\Pr\left[\exists k \in [t], i \in [\ell] : \left|X^j_{i,k} \setminus \left(\bigcap_{m=1}^{\ell} X^j_{m,k}\right)\right| > 1 + 3\ln(2t\ell)\right] \leq 1/2.$$

It follows that

$$\Pr\left[\exists \begin{array}{c} k_1,\dots k_\epsilon \in [t] \\ i_1,\dots,i_\epsilon \in [\ell] \end{array} : \left|X^j_{i_j,k_j} \setminus \bigcap_{m=1}^{\ell} X^j_{m,k_j}\right| \geq 1+3\ln(2t\ell)\ \forall j \in [\epsilon]\right]$$
$$= \prod_{j=1}^{\epsilon} \Pr\left[\exists k_j \in [t], i_j \in [\ell] : \left|X^j_{i_j,k_j} \setminus \bigcap_{m=1}^{\ell} X^j_{m,k_j}\right| \geq 1+3\ln(2t\ell)\ \forall j \in [\epsilon]\right] \leq 2^{-\epsilon}.$$

□

Corollary 4. *Let $n, t, \tilde{t} \in \mathbb{N}$ with $t < n$ and $\tilde{t} \geq 1+3\ln(2t\ell)$. For each $j \in [\epsilon]$, let $H^j : U \to [t]$ be a random function. Let $X_1,\dots,X_\ell \in U^n$ be sets of size n and $(X^j_{i,1},\dots,X^j_{i,t}) \leftarrow H^j(X_i)$ for $j \in [\epsilon]$ and $i \in [\ell]$. If there exist indices $k_1,\dots k_\epsilon \in [t]$ and $i_1,\dots,i_\epsilon \in [\ell]$, such that for all $j \in [\epsilon]$ it holds that*

$$\left|X^j_{i_j,k_j} \setminus \bigcap_{m=1}^{\ell} X^j_{m,k_j}\right| \geq \tilde{t},$$

then

$$\Pr\left[\left|\bigcap_{i=1}^{\ell} X_i\right| < n-t\right] \geq 1-2^{-\epsilon}$$

where the probability is taken over the random choices of $H^1,\dots H^\epsilon$.

When partitioning a set into several subsets randomly, one cannot guarantee that all subsets will be of the same size. This is problematic, since we would like to view each party's buckets as inputs to smaller instances of a private multiparty intersection cardinality testing problem. That is, if the different parties have inputs of different (secret) sizes, then it is not clear what it means for an intersection to be large enough. For this reason, each party will not directly input its subset, but rather a padded version of it. Since the communication complexities of our protocols never depend on the input sizes and since we only care about the asymptotic communication complexity, we simply pad each bucket to its maximum size.

Lemma 5. *Let $n, \ell, t, \tilde{t}, b \in \mathbb{N}$ with $t \leq \tilde{t} < n$ and $b := n$ and $H : U \to [t]$ be a random function. Let $X_1,\dots,X_\ell \in U^n$ be sets of size n and $(X_{i,1},\dots,X_{i,t}) \leftarrow H(X_i)$ for $i \in [\ell]$. If*

$$\left|X_{j,k} \setminus \bigcap_{i=1}^{\ell} X_{i,k}\right| < \tilde{t}, \forall j \in [\ell], k \in [t], \tilde{t} \in \mathbb{N}$$

then

$$\left|\mathtt{Pad}(X_{j,k},b) \setminus \bigcap_{i=1}^{\ell} \mathtt{Pad}(X_{i,k},b)\right| < \tilde{t}, \forall j \in [\ell], k \in [t].$$

Proof. Fix some $k \in [t]$ and define $t_j := \left|X_{j,k} \setminus \bigcap_{i=1}^{\ell} X_{i,k}\right|$ for $j \in [\ell]$. Observe that

$$t_j + \left|\bigcap_{i=1}^{\ell} X_{i,k}\right| + |\mathtt{Pad}(X_{j,k},b) \setminus X_{j,k}| = b$$

and thus for $j, j' \in [\ell]$ we have

$$t_j + \left|\bigcap_{i=1}^{\ell} X_{i,k}\right| + |\mathtt{Pad}(X_{j,k},b) \setminus X_{j,k}| = t_{j'} + \left|\bigcap_{i=1}^{\ell} X_{i,k}\right| + |\mathtt{Pad}(X_{j',k},b) \setminus X_{j',k}|$$

$$\iff t_j + |\mathtt{Pad}(X_{j,k}, b) \setminus X_{j,k}| - |\mathtt{Pad}(X_{j',k}, b) \setminus X_{j',k}| = t_{j'}$$

Now consider index j' such that $|X_{j',k}| \geq |X_{j,k}|$ for any other $j \in [\ell]$. For that index j' it holds that $\mathtt{Pad}(X_{j',k}, b) \setminus X_{j',k} \subseteq \mathtt{Pad}(X_{j,k}, b) \setminus X_{j,k}$. In other words, this means that the elements that were used for padding the bucket belonging to party j' will be exactly the added elements in the intersection. Thus, for any other j the number of elements not in the intersection will be $t_j + |\mathtt{Pad}(X_{j,k}, b) \setminus X_{j,k}| - |\mathtt{Pad}(X_{j',k}, b) \setminus X_{j',k}|$. Now if by assumption $t_{j'} < \tilde{t}$, then $t_j + |\mathtt{Pad}(X_{j,k}, b) \setminus X_{j,k}| - |\mathtt{Pad}(X_{j',k}, b) \setminus X_{j',k}| < \tilde{t}$. □

Combining all of the above observations, we now get the following lemma.

Theorem 6. *Let $n, \ell, t, \tilde{t}, b \in \mathbb{N}$ with $t < n$, $\tilde{t} \geq 1 + 3\ln(2t\ell)$ and let $b = n$. For each $j \in [\epsilon]$, let $H^j : U \to [t]$ be a random function. Let $X_1, \ldots, X_\ell \in U^n$ be sets of size n and $(X^j_{i,1}, \ldots, X^j_{i,t}) \leftarrow H^j(X_i)$ for $j \in [\epsilon]$ and $i \in [\ell]$. If*

$$\left| \bigcap_{i=1}^{\ell} X_i \right| \geq n - t,$$

then

$$\Pr\left[\forall k \in [t], \forall i \in [\ell], \exists j \in [\epsilon] : \left| \mathtt{Pad}(X^j_{i,k}, b) \setminus \bigcap_{m=1}^{\ell} \mathtt{Pad}(X^j_{m,k}, b) \right| < \tilde{t} \right] \geq 1 - 2^{-\epsilon},$$

where the probability is taken over the random choice of $H^1, \ldots, H^\epsilon$.

Proof. By Theorem 5 we know that if $\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n - t$, then for all $i \in [\ell]$ and $k \in [t]$, there exist $j \in [\epsilon]$, such that $\left|X^j_{i,k} \setminus \bigcap_{m=1}^{\ell} X^j_{m,k}\right| < \tilde{t}$ with overwhelming probability. Also from Lemma 5 we know $\left|\mathtt{Pad}(X^j_{i,k}, b) \setminus \bigcap_{m=1}^{\ell} \mathtt{Pad}(X^j_{m,k}, b)\right| < \tilde{t}$ in that case. The proof directly follows from these two observations. □

Functionality $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}(X_1, \ldots, X_\ell)$

$u := \left| \bigcap_{i=1}^{\ell} X_i \right|$

if $u < n - t$

  return $\mathtt{Share}_n(v)$

else

  return $\mathtt{Share}_n(u)$

Fig. 10. Functionality for computing the number of elements in the intersection.

Armed with the insights from above, we are now ready to present our multiparty construction. We will assume that we are given access to an ideal functionality $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}$ as depicted in Figure 10 and we will show how to concretely instantiate it in Section 4.1. We also use two other simple functionalities $\mathcal{F}_{\ell\text{-min}}$ and $\mathcal{F}^{n,t,\epsilon}_{\ell\text{-vec-intrvl}}$ in our protocol, which are described in Figure 4 and Figure 5 respectively. Note that these functionalities can be implemented using any generic MPC protocol with communication complexities that are independent of the initial set size n or threshold t.

In Figure 11 we instantiate the protocol for multiparty private cardinality testing. Similar to the two-party case, here all the parties throw their set elements into t buckets and then run separate instances of cardinality test protocol among those buckets with a threshold parameter $\tilde{t}$, as stated in Theorem 6.

Construction $\Pi^{n,t}_{\ell\text{-pict}}(X_1, \ldots, X_\ell)$

for $j \in [\epsilon]$:
  for $i \in [\ell]$:
    Party i **:**
    $(X^j_{i,1}, \ldots, X^j_{i,t}) \leftarrow H^j(X_i)$
  for $k \in [t]$:
    $(s^j_{1,k}, \ldots, s^j_{\ell,k}) \leftarrow \mathcal{F}^{n,\tilde{t},n+1}_{\ell\text{-}\cap}\left(\mathtt{Pad}(X^j_{1,k}, n+1), \ldots, \mathtt{Pad}(X^j_{\ell,k}, n+1)\right)$
    $(d^j_{1,k}, \ldots, d^j_{\ell,k}) \leftarrow \mathcal{F}_{\ell\text{-min}}\left(\{\left|\mathtt{Pad}(X^j_{i,k}, n+1)\right| - \left|X^j_{i,k}\right|\}_{i\in[\ell]}\right)$
  for $i \in [\ell]$:
    Party i **:** $s^j_i = \sum_{k=1}^{t}(s^j_{i,k} - d^j_{i,k})$
$(r_1, \ldots, r_\ell) \leftarrow \mathcal{F}^{n,t,\epsilon}_{\ell\text{-vec-intrvl}}((s^1_1, \ldots, s^\epsilon_1), \ldots, (s^1_\ell, \ldots, s^\epsilon_\ell))$
return $(r_1, \ldots, r_\ell)$

Fig. 11. Protocol for multiparty private intersection cardinality testing.

Theorem 7. *Let $n, t, \tilde{t} \in \mathbb{N}$ with $n > t$ and $\tilde{t} \geq 1 + 3\ln(2t\ell)$. The protocol $\Pi_{\ell\text{-pict}}$ depicted in Figure 11 securely realizes $\mathcal{F}^{n,t}_{\ell\text{-pict}}$ using $\epsilon \cdot t$ calls to $\mathcal{F}^{b,\tilde{t},n+1}_{\ell\text{-}\cap}$ and $\mathcal{F}_{\ell\text{-min}}$ and one call to $\mathcal{F}^{n,t,\epsilon}_{\ell\text{-vec-intrvl}}$.*

Proof. We prove correctness and privacy separately.

Correctness. If $\left|\bigcap_{i=1}^{\ell} X_i\right| < n - t$, then by Theorem 6, we know that no bucket will overflow with an overwhelming probability in which case the protocol computes the size of the intersection correctly. If $\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n-t$, then two things can happen. Either there will exist indices $k_1, \ldots k_\epsilon \in [t]$ and $i_1, \ldots, i_\epsilon \in [\ell]$, such that for all $j \in [\epsilon]$ it holds that $\left|X^j_{i_j,k_j} \setminus \bigcap_{m=1}^{\ell} X^j_{m,k_j}\right| \geq \tilde{t}$. In this case, by Corollary 4, we know that the intersection was too small with an overwhelming probability. By construction, each sum of secret shared values per partitioning will contain a summand of $n+1$ and thus the sums will always be larger than n in which case $\mathcal{F}^{n,t,\epsilon}_{\ell\text{-vec-intrvl}}$ returns 0 as desired.

Otherwise, the intersection is too small, but no bucket overflows. Since no bucket overflows, the parties correctly compute a secret sharing of the actual intersection and thus $\mathcal{F}^{n,t,\epsilon}_{\ell\text{-vec-intrvl}}$ will produce the correct output of the computation.

Privacy. Without loss of generality assume that $P_1, \ldots, P_{\ell-1}$ are corrupted. We observe that the only communication the parties have during a protocol execution is through oracle calls. Each oracle call returns a fresh secret sharing of some random value and the parties always receives a subset of shares that is insufficient to reconstruct. To simulate the corrupted parties' views, we simply return shares of fresh secret sharings of 0 for each oracle call.

□

4.1 Instantiating $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}$

To instantiate $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}$, we use $\mathcal{F}^{n,i}_{\ell\text{-pict}}$ once for each threshold $i \in [t]$ and then accumulate the result. We also use $\mathcal{F}^{n-t,v}_{\ell\text{-geq}}$ functionality, described in Section 2, which checks whether the secret shared values obtained from $\mathcal{F}^{n,i}_{\ell\text{-pict}}$ indicates that the size of the intersection is greater than $n - t$. If that is the case $\mathcal{F}_{\ell\text{-geq}}$ returns the

Construction $\Pi^{n,t,v}_{\ell\text{-}\cap}(X_1, \ldots, X_\ell)$

1 : **for** $i \in [t]$

2 : $\quad (r_{1,i}, \ldots, r_{\ell,i}) \leftarrow \mathcal{F}^{n,i}_{\ell\text{-pict}}(X_1, \ldots, X_\ell)$

3 : **for** $i \in [\ell]$

4 : $\quad$ **Party** i **:** $r_i := \sum_{j=1}^{t} r_{i,j}$

5 : **Party** 1 **:** $r_1 := n - t - 1 + r_1$

6 : $(d_1, \ldots, d_\ell) \leftarrow \mathcal{F}^{n-t,v}_{\ell\text{-geq}}(r_1, \ldots, r_\ell)$

7 : **return** $(d_1, \ldots, d_\ell)$

Fig. 12. Protocol $\Pi^{n,t,v}_{\ell\text{-}\cap}$ realizing $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}$.

exact size of the intersection, otherwise it returns the default value v. The protocol $\Pi^{n,t,v}_{\ell\text{-}\cap}$ is described in Figure 12.

Theorem 8. *Let $n, t \in \mathbb{N}$ with $n > t$ and $v \in \mathbb{F}$. The protocol $\Pi^{n,t,v}_{\ell\text{-}\cap}$ depicted in Figure 12 securely implements $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}$ using one call to $\mathcal{F}^{n,i}_{\ell\text{-pict}}$ for each $i \in [t]$ and one call to $\mathcal{F}^{n-t,v}_{\ell\text{-geq}}$.*

Proof. For correctness, we observe that $\mathcal{F}^{n,i}_{\ell\text{-pict}}(X_1, \ldots, X_\ell)$ returns a sharing of 1, whenever $\left|\bigcap_{i=1}^{\ell} X_i\right| \geq n-i$. Thus, if $\left|\bigcap_{j=1}^{\ell} X_j\right| \geq n - t$, then $\mathcal{F}_{\ell\text{-pict}}$ will return sharing of 1 exactly $t - t^* + 1$ times, where $n - t^*$ is the true intersection size. Consequently the protocol returns a secret sharing of $n-t-1+(t-t^*+1) = n-t^* = \left|\bigcap_{j=1}^{\ell} X_j\right|$.

Privacy. Without loss of generality assume that $P_1, \ldots, P_{\ell-1}$ are corrupted. The view of the corrupted parties only contain received messages from the oracles. Each oracle query to $\mathcal{F}^{n,i}_{\ell\text{-pict}}$ returns a fresh secret sharing, which can be simulated by providing the corrupted parties with fresh shares of secret sharings of 0. The last query to $\mathcal{F}^{n-t,v}_{\ell\text{-geq}}(r_1, \ldots, r_\ell)$ can be simulated by returning the outputs given to the simulator. Indistinguishability of the simulated transcript from the real one directly follows from the security guarantees of additive secret sharing.

□

We can use the protocol of Branco, Döttling, and Pu [BDP21] to instantiate $\mathcal{F}^{n,t}_{\ell\text{-pict}}$ in $\mathcal{F}^{n,t,v}_{\ell\text{-}\cap}$. They present an ℓ-party protocol with a communication complexity of $\mathcal{O}(\lambda \ell t^2 \operatorname{polylog} t)$ bits based on additively homomorphic encryption. Their protocol can easily be extended to use generic secure computation techniques in all places, where additively homomorphic encryption was used. With this change, their protocol provides a solution based on, for instance, oblivious transfer with a communication complexity of $\mathcal{O}(\lambda \ell \operatorname{poly}(t))$ bits.

In our instantiation, we have ϵt buckets and for each of them we execute the protocol of Branco et al. $\mathcal{O}(\ln t)$ times with a threshold of $\mathcal{O}(\ln t)$. Thus we get a total communication complexity of $\mathcal{O}(\epsilon \lambda \ell t \operatorname{polylog} t)$.

Corollary 5. *Assuming the existence of oblivious transfer and or additively homomorphic encryption, there exists a protocol for securely computing the ℓ-party private intersection cardinality test for threshold t with communication complexity of $\mathcal{O}(\epsilon^2 \ell t \operatorname{polylog} t)$ bits.*

Combining the results in our paper with the protocols for actually computing the intersection, once it is known that it is large enough, from by Ghosh and Simkin [GS19], we get the following result.

Corollary 6. *Assuming the existence of oblivious transfer or additively homomorphic encryption, there exists a passively secure protocol for threshold private set intersection among ℓ parties with threshold t with communication complexity of $\mathcal{O}(\epsilon^2 \ell t \operatorname{polylog} t)$ bits.*

References

BDP21. Pedro Branco, Nico Döttling, and Sihang Pu. Multiparty cardinality testing for threshold private intersection. In Juan Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12711 of *Lecture Notes in Computer Science*, pages 32–60, Virtual Event, May 10–13, 2021. Springer, Heidelberg, Germany.

BMRR21. Saikrishna Badrinarayanan, Peihan Miao, Srinivasan Raghuraman, and Peter Rindal. Multi-party threshold private set intersection with sublinear communication. In Juan Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12711 of *Lecture Notes in Computer Science*, pages 349–379, Virtual Event, May 10–13, 2021. Springer, Heidelberg, Germany.

CD01. Ronald Cramer and Ivan Damgård. Secure distributed linear algebra in a constant number of rounds. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 119–136, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

DCW13. Changyu Dong, Liqun Chen, and Zikai Wen. When private set intersection meets big data: an efficient and scalable protocol. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013: 20th Conference on Computer and Communications Security*, pages 789–800, Berlin, Germany, November 4–8, 2013. ACM Press.

DPT20. Thai Duong, Duong Hieu Phan, and Ni Trieu. Catalic: Delegated PSI cardinality with applications to contact tracing. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Part III*, volume 12493 of *Lecture Notes in Computer Science*, pages 870–899, Daejeon, South Korea, December 7–11, 2020. Springer, Heidelberg, Germany.

FNP04. Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

GS19. Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 3–29, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

KKRT16. Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious PRF with applications to private set intersection. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 818–829, Vienna, Austria, October 24–28, 2016. ACM Press.

KS05. Lea Kissner and Dawn Xiaodong Song. Privacy-preserving set operations. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 241–257, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Heidelberg, Germany.

Lin17. Yehuda Lindell. How to simulate it–a tutorial on the simulation proof technique. *Tutorials on the Foundations of Cryptography*, pages 277–346, 2017.

Mar14. Moxie Marlinspike. The difficulty of private contact discovery. whispersystems.org/blog/contact-discovery., 2014.

Mea86. Catherine A. Meadows. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In *Proceedings of the 1986 IEEE Symposium on Security and Privacy, Oakland, California, USA, April 7-9, 1986*, pages 134–137, 1986.

NMH+10. Shishir Nagaraja, Prateek Mittal, Chi-Yao Hong, Matthew Caesar, and Nikita Borisov. Botgrep: Finding P2P bots with structured graph analysis. In *19th USENIX Security Symposium, Washington, DC, USA, August 11-13, 2010, Proceedings*, pages 95–110, 2010.

PRTY19. Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. SpOT-light: Lightweight private set intersection from sparse OT extension. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 401–431, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

PRTY20. Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. PSI from PaXoS: Fast, malicious private set intersection. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 739–767, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

PSSZ15. Benny Pinkas, Thomas Schneider, Gil Segev, and Michael Zohner. Phasing: Private set intersection using permutation-based hashing. In *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015.*, pages 515–530, 2015.