On Secure Multiparty Sampling For More than Two Parties

Manoj M. Prabhakaran
UIUC, USA
Email: mmp@illinois.edu

Vinod M. Prabhakaran
TIFR, India
Email: vinodmp@tifr.res.in

***Abstract*—We investigate secure multi-party sampling problems involving more than two parties. In the public discussion model, we give a simple characterization of the distributions that can be sampled without any setup. In a model which allows private point-to-point communication, we reduce the problem of characterizing distributions that can be securely sampled using pairwise setups to the problem of characterizing distributions that can be sampled without any setups.**

I. Introduction

Secure multi-party computation is the following problem (see Figure 1): a set of parties $P_1, \ldots, P_m$ are given inputs $X_1, \ldots, X_m$ respectively, and are required to output $Y_1, \ldots, Y_m$ respectively, according to a pre-specified distribution $p_{Y_1,\ldots,Y_m|X_1,\ldots,X_m}$. The security requirement (against passive (a.k.a. semi-honest) adversaries) is that, any subset of (corrupt) parties $C \subseteq \{1, \ldots, m\}$ cannot infer (from what they receive during the protocol) anything more about the inputs and outputs of the other parties. The parties communicate with each other using public discussion (or private point-to-point channels, depending on the model). To aid them in this process, the parties are given a *setup*: additional inputs $S_1, \ldots, S_m$ respectively, drawn from a joint distribution (independent of $X_1, \ldots, X_m$). Both the question of feasibility (does there exist a protocol for securely computing $p_{Y_1,\ldots,Y_m|X_1,\ldots,X_m}$ using a setup $p_{S_1,\ldots,S_m}$) and the question of efficiency (how many copies of the setup are required per instance of the output, and using how much communication) are of interest.

In this paper we consider a special case of the above problem, in which there are no inputs. It is well-known in the 2-party case, that most "interesting" distributions can be sampled only when there is a "non-trivial" setup (see, for instance, [1] and references therein). We seek to understand the power of various setups in the multi-party case. We restrict ourselves to the feasibility question.

Our first result considers the point-to-point channels model. We investigate the power of "pairwise setups," (see Figure 2), and show that the distributions that can be sampled with such setups are exactly those for which a related distribution can be sampled without any setups. We point out that it remains open to characterize which distributions can be securely sampled without setups. Nevertheless, our result *reduces* the problem of characterizing distributions that can be sampled using pairwise setups to the problem of characterizing distributions that can be sampled without any setups. We present the solution for the latter problem in the case of 3 parties, but leave it open for more than 3 parties.

Our second result considers the public discussion model. Here we give a simple characterization of the distributions that can be sampled without any setup.

Related Work: There has been a large body of work on secure multiparty computation (see for instance, [2] and references therein). In particular, the set of (randomized) functions that are computable using various functions as setups, have been fully or partially characterized in various settings. Below we mention a few such results, restricted to the setting without computational restrictions, and with security only against passive corruption. For comparison, note that sampling refers to evaluating *randomized functions without input*, and our setups are also inputless.

Two-party functions (deterministic or randomized, with or without inputs) that are "complete" as setups — i.e., that can be used to evaluate any function — were characterized in [3]. The commonly considered setups in the setting with more than two parties typically consists of independent instances of a two-party setup between every pair of parties. But other kinds of setups have also been considered. For instance, in [4], it is shown that a setup composed of independent instances of two-party setups between a strict subset of all pairs is not complete. (This is a consequence of our first result as well.) In [5] setups composed of independent instances of k-party setups are considered for various values of k.

There have also been many results on what can be securely computed without any setups. In the two-party setting, *deterministic* functions (with inputs) that can be securely evaluated without setups was characterized in [6], [7], [8]. For the public discussion model, this characterization of deterministic functions extends to more than two parties as well [7]. Our result in the public discussion model complements this characterization (for deterministic functions with inputs) with a characterization of securely realizable *randomized* functions *without inputs*; this leaves the characterization open for randomized functions with inputs (which is open even for two-parties). In the private-channels model, for more than two parties, it remains open to characterize deterministic functions or randomized functions, with or without inputs, that are securely realizable without setups. Several negative results can be derived based on negative results in the two-party setting

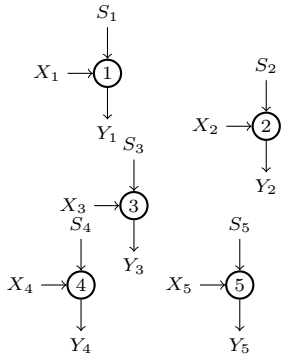

Figure 1. The secure multiparty computation (MPC) problem. A network of nodes with private inputs $(x_1, \ldots, x_N)$ want to "securely compute" dependent outputs Y_i satisfying a conditional joint distribution $p(y_1, \ldots, y_N | x_1, \ldots, x_N)$. The resources available are setup random variables S_i which are independent of the inputs, and a communication network of noiseless links between every pair of nodes.

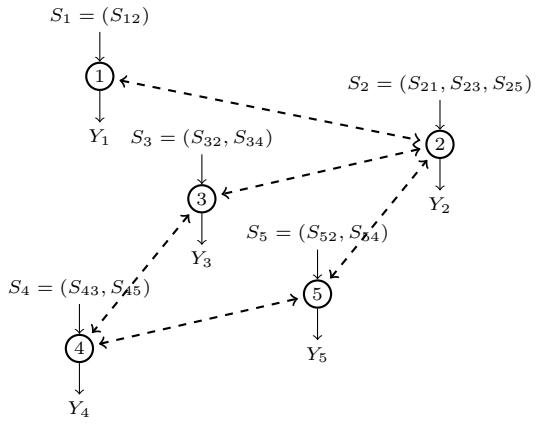

Figure 2. The secure sampling problem studied in the first part of this paper is a special case of secure MPC. There are no inputs, and the nodes want to securely sample from a joint distribution $p(y_1, \ldots, y_N)$. A broken line between nodes i and j represents an independent non-trivial pair S_{ij}, S_{ji} available to the two nodes as part of the setup. The communication graph (not shown) is complete. Parties communicate over noiseless pairwise private links. In the problem studied in the second part of this paper (in Section VI), the communication is over a noiseless public discussion channel and there are no setup variables.

using a partitioning argument, but [9] showed that there exist more negative results than can be deduced thus.

As mentioned above, the sampling problem is a special case of the secure multi-party computation problem, which has recently received some attention in the Information Theory community [1], [10].

II. Problem formulation

Consider a desired joint distribution (p.m.f.) $p_{Y_1,\ldots,Y_m}$ defined over a finite alphabet $\mathcal{Y}_1 \times \ldots \times \mathcal{Y}_m$. We will denote this by $p_{\mathbf{Y}}$. The goal of secure sampling is for m parties to engage in a protocol at the end of which the i-th party produces an output $\widehat{Y}_i \in \mathcal{Y}_i$ such that (a) the joint distribution of $(\widehat{Y}_1, \ldots, \widehat{Y}_m)$ is close to the desired joint distribution and (b) any subset of parties has no further knowledge of the rest of the parties' outputs than they can infer from their own output. We will make these more precise below. The parties have access to a setup: independent copies of jointly distributed random variables with distribution $p_{S_1,\ldots,S_m}$, with party-i having access to copies of S_i. We will denote the setup joint distribution by $p_{\mathbf{S}}$ and the output joint distribution by $p_{\widehat{\mathbf{Y}}}$. In addition they also have access to independent private randomness. The protocol the parties engage in has some finite number of rounds. In each round, each pair of parties exchange private messages (binary strings) with each other over a bidirectional private noiseless channel connecting the pair. The messages party-i sends is a function of its setup random variables (copies of S_i), its own private randomness, and all the messages it received in all the previous rounds. We will call the set of all random variables available to a party-i at the end of a protocol Π as its *view*, Π_i^{view}. This consists of its setup random variable, private randomness, and all the messages it received during the protocol. At the end of the protocol, party-i outputs $\widehat{Y}_i = \Pi_i^{\text{out}}$ which is a function of its view Π_i^{view}.

We will use the following notation: $[m] = \{1, \ldots, m\}$. If $Z_i,\ i \in [m]$ are random variables and $T \subseteq [m]$, we use Z_T to denote the collection of random variables $Z_i,\ i \in T$.

We denote the above sampling problem by $\text{Sam}_{\text{pvt}}(p_{\mathbf{Y}}|p_{\mathbf{S}})$, where pvt denotes the fact that the parties communicate over pairwise *private* channels. In Section VI we will consider the case of communication over *public* discussion channels. When there is no setup, we denote the problem by $\text{Sam}_{\text{pvt}}(p_{\mathbf{Y}})$.

Definition 1. *For $\epsilon \geq 0$, a protocol Π is said to* ϵ-securely realize $\text{Sam}_{\text{pvt}}(p_{\mathbf{Y}}|p_{\mathbf{S}})$ *if, for each $C \subseteq [m]$ there exists a random variable ("simulated view") Σ_C^{view} (jointly distributed with the ideal outputs $Y_{[m]}$), such that*

$$\Sigma_C^{\text{view}} - Y_C - Y_{\overline{C}} \tag{1}$$

$$\Delta\big(\ \big(Y_{\overline{C}}, \Sigma_C^{\text{view}}\big), \big(\Pi_{\overline{C}}^{\text{out}}, \Pi_C^{\text{view}}\big)\ \big) \leq \epsilon \tag{2}$$

Here $\Delta(\cdot,\cdot)$ stands for the total variation distance. In this case we say $\Pi^{\mathbf{S}} \overset{\epsilon}{\rightsquigarrow} \mathbf{Y}$, where the superscript $\mathbf{S}$ denotes the setup. When there is no setup, we say $\Pi \overset{\epsilon}{\rightsquigarrow} \mathbf{Y}$.

When $\epsilon = 0$, the security obtained is called *perfect security*. Note that in the first condition we can write $Y_{[m]}$ instead of $Y_{\overline{C}}$. We provide the intuition behind this definition in Section III.

We say that $\text{Sam}_{\text{pvt}}(p_{\mathbf{Y}}|p_{\mathbf{S}})$ is *securely realizable* if for every $\epsilon > 0$, there is a protocol Π such that $\Pi^{\mathbf{S}} \overset{\epsilon}{\rightsquigarrow} \mathbf{Y}$.

III. Preliminaries

All the random variables considered in this paper have finite alphabets. As mentioned before $\Delta(X, Y)$ would stand for the total variation distance between random variables X and Y. For clarity, sometimes we write this in terms of the probability distributions of the random variables, as $\Delta(p_X, p_Y)$.

In the case of secure 2-party sampling, a distribution p_{Y_1,Y_2} is called *trivial* if there is a random variable Q defined by $p_{Q|Y_1Y_2}$ such that $H(Q|Y_1) = H(Q|Y_2) = 0$ and $I(Y_1; Y_2|Q) = 0$. It is well-known that only trivial 2-party distributions are securely realizable without any setup (see,

for instance, [1]). A joint distribution which is not trivial is called *non-trivial*.

Oblivious Transfer (or OT) is an important non-trivial 2-party distribution defined as follows (there are several equivalent formulations): $Y_1 = (\alpha, \alpha')$, $Y_2 = (\beta, \beta')$ where $\alpha, \beta, \alpha', \beta'$ are random variables uniformly distributed over elements of $\{0,1\}^4$ which satisfy $\alpha\beta = \alpha' \oplus \beta'$.

Given OT as a setup, *any* two-party computation is securely realizable.[1] This is a consequence of a protocol in [11] which (including simplifications from [13], [14]) we shall refer to as the *Basic-GMW protocol*. The Basic-GMW protocol can in fact support a "reactive" functionality, as explained later. In the full version, we include a brief self-contained sketch of the Basic-GMW protocol.

Explanation of the security definition: The security definition can be thought of as comparing the actual protocol execution with an "ideal" scenario where the outputs are sampled by a trusted outsider and handed over to the parties. In either scenario, an adversary is allowed to (passively) corrupt any arbitrary subset of players and learn the outputs of these players. This is the set C in the definition. The net effect or outcome of the execution (of the real protocol or in the ideal scenario) consists of the outputs of all the parties and the view reported by the adversary (which, in the real execution, includes the internal states of all the corrupt parties). In the ideal execution the adversary sees only the outputs obtained by the corrupt players from the trusted party; however we allow the adversary to report a "simulated" view as if it took part in the protocol. A protocol is secure if, for an adversary corrupting any subset C of the players, there exists such a simulator (for the set C), such that the outcome of the actual protocol execution and the outcome of the execution in the ideal scenario are (almost) identically distributed. The first item in the security definition ensures that the simulated view can indeed be generated in the ideal scenario — i.e., based on the outputs of the corrupt parties alone (since this is the view of the adversary in the ideal scenario). The second item states that the outcome of the ideal execution is not very differently distributed than the outcome of the actual protocol execution. Intuitively, this means that whatever can happen in the real protocol execution could have happened in the ideal scenario too (after corrupting the same set of players), and since the latter is "secure" by design (or rather, it sets the security goal) the former should be considered secure too. In particular, note that when $C = \emptyset$, the second item requires that the outputs produced by the protocol is distributed close to the desired distribution.

If we require perfect security, then the security definition implies that the output distribution of the protocol is exactly $p_{\mathbf{Y}}$ (by considering say, $C = \emptyset$) and that the view of the adversary in the protocol gives absolutely no information about the outputs of the honest parties other than what is implied by the corrupt parties' outputs alone.

We remark that the typical definition in cryptography literature requires the protocol to be *uniform* (i.e., can be implemented by a single Turing Machine that takes ϵ as input) and also "efficient" (i.e., the Turing Machine implementing the protocol runs in time (say) polynomial in $\log 1/\epsilon$). Our results continue to hold with such a definition (because, in particular, the protocols we use as well as the modifications we make to given protocols are all uniform); however, for simplicity we omit these requirements.

[1]This readily extends to m-party computation as well [11], and further, it holds when security against active adversaries is considered [12]. [11] also offers security against active adversaries, in a weaker model in which the setup can be "implemented" from scratch, by relying on computational complexity assumptions. We do not require these results.

IV. Reduction to secure sampling without setup

Definition 2. *$\mathfrak{P}^m$ is the set of all joint distributions (p.m.f.'s) of m finite-alphabet random variables. $\mathfrak{P}_2^m \subset \mathfrak{P}^m$ are the set of all joint distributions $p_{S_1,\ldots,S_m}$ where each S_i can be written as the collection $\{(S_{ij}, j)\}_{\{i,j\}\in\mathcal{E}}$, where $\mathcal{E}$ is the set of edges of a simple graph $G([m], \mathcal{E})$, whose vertices are $[m] = \{1, \ldots, m\}$, and for each $\{i,j\} \in \mathcal{E}$, there is a pair of random variables (S_{ij}, S_{ji}) and these pairs are independent across edges, i.e.,*

$$p_{\cup_{\{i,j\}\in\mathcal{E}}\{S_{ij},S_{ji}\}} = \prod_{\{i,j\}\in\mathcal{E}} p_{S_{ij},S_{ji}}.$$

See Figure 2 for an example. Note that the two random variables in a pair (S_{ij}, S_{ji}) are jointly distributed, but different pairs are independent of each other.

Definition 3. *For $p_{S_1,\ldots,S_m} \in \mathfrak{P}_2^m$, we define $\mathcal{E}(p_{S_1,\ldots,S_m}) \subset \mathcal{E}$ (where $\mathcal{E}$ is as above) such that $\{i,j\} \in \mathcal{E}(p_{S_1,\ldots,S_m})$ if the pair of random variables (S_{ij}, S_{ji}) is non-trivial.*

Definition 4. *If $\mathcal{P} = \mathrm{Sam}_{\mathrm{pvt}}(p_{\mathbf{Y}}|p_{\mathbf{S}})$ where $p_{\mathbf{Y}} \in \mathfrak{P}^m$ is the joint distribution of a set of m random variables $\{Y_i | i \in [m]\}$ and $p_{\mathbf{S}} \in \mathfrak{P}_2^m$, then the sampling problem $\mathrm{Flat}(\mathcal{P})$ is defined to be $\mathrm{Sam}_{\mathrm{pvt}}(p_{\mathbf{Y}} \times p_{\mathbf{Z}|\mathbf{Y}})$ (i.e., sample $p_{\mathbf{Y}} \times p_{\mathbf{Z}|\mathbf{Y}}$ without any setups) where $\mathbf{Z}$ is a set of random variables $\{Z_{\{i,j\}} | \{i,j\} \in \mathcal{E}(p_{\mathbf{S}})\}$, such that $Z_{\{i,j\}} = (Y_i, Y_j)$.*

Note that $\mathrm{Flat}(\mathcal{P})$ is a sampling problem for parties $[m] \cup \mathcal{E}(p_{\mathbf{S}})$ with no setups.

Theorem 1. *Let $\mathcal{P} = \mathrm{Sam}_{\mathrm{pvt}}(p_{\mathbf{Y}}|p_{\mathbf{S}})$ where $p_{\mathbf{Y}} \in \mathfrak{P}^m$ and $p_{\mathbf{S}} \in \mathfrak{P}_2^m$. Then $\mathcal{P}$ is securely realizable if and only if $\mathrm{Flat}(\mathcal{P})$ is securely realizable.*

Proof: We shall present two constructions, one for each direction in the statement. Due to lack of space, we omit the proofs of security of these constructions (see full version).

Construction 1 ("only if"): Firstly, if $\Pi^{\mathbf{S}} \overset{\epsilon}{\rightsquigarrow} \mathbf{Y}$, then we give a protocol $\widetilde{\Pi}$ such that $\widetilde{\Pi} \overset{\epsilon'}{\rightsquigarrow} (\mathbf{Y}, \mathbf{Z})$ where $\epsilon' \downarrow 0$ and $\epsilon \downarrow 0$. We denote the parties in $\widetilde{\Pi}$ that output Y_i, $i \in [m]$, by P_i and the parties that output $Z_{\{i,j\}}$, $\{i,j\} \in \mathcal{E}(p_{\mathbf{S}})$ by $\widetilde{P}_{\{i,j\}}$. The protocol $\widetilde{\Pi}$ proceeds as follows:

1) For each $\{i,j\} \in \mathcal{E}(p_{\mathbf{S}})$, the party $\widetilde{P}_{\{i,j\}}$ generates (using its private randomness) and supplies S_{ij} to P_i and S_{ji} to P_j. For $\{i,j\} \notin \mathcal{E}(p_{\mathbf{S}})$, (S_{ij}, S_{ji}) is trivial, and the

parties P_i and P_j generate S_{ij} and S_{ji} using their private randomness and possibly one message between them.
2) The parties $P_1, \ldots, P_m$ carry out an execution of Π, but using S_{ij} obtained as above as the setup.
3) At the end of this execution, P_i and P_j send their outputs Y_i and Y_j to $\widetilde{P}_{\{i,j\}}$, who outputs Y_i, Y_j.

The proof of security of this protocol follows from the security of Π, and the fact that $\widetilde{P}_{\{i,j\}}$ is allowed to learn the outputs of P_i and P_j: if $\widetilde{P}_{\{i,j\}}$ is corrupt one might as well consider both P_i and P_j to be corrupt, and then the view of the adversary in an execution of $\widetilde{\Pi}$ can be identified with the view of the adversary in an execution of Π.

Construction 2 ("if"): Now, we consider the other direction: if $\mathrm{Flat}(\mathcal{P})$ is securely realizable, then so is $\mathcal{P}$. Suppose $\widetilde{\Pi} \overset{\epsilon}{\rightsquigarrow} (\mathbf{Y}, \mathbf{Z})$; then we shall give a protocol Π such that $\Pi^{\mathbf{S}} \overset{\delta}{\rightsquigarrow} \mathbf{Y}$ where δ is such that $\delta \downarrow 0$ if $\epsilon \downarrow 0$. We do this in two steps:

1) First we modify the protocol $\widetilde{\Pi}$ to $\widehat{\Pi}$, so that $\widehat{\Pi} \overset{\epsilon}{\rightsquigarrow} (\mathbf{Y}, \mathbf{Z})$, and in $\widehat{\Pi}$ each party of the form $\widetilde{P}_{\{i,j\}}$ communicates only with the two parties P_i and P_j.
2) Next, we shall eliminate the parties $\widetilde{P}_{\{i,j\}}$ in $\widehat{\Pi}$ to obtain Π.

$\widehat{\Pi}$ is identical to $\widetilde{\Pi}$ except for the communication channels for parties of the form $\widetilde{P}_{\{i,j\}}$. For communication with (from or to) any party other than P_i and P_j, $\widetilde{P}_{\{i,j\}}$ uses the channel split_{ij} defined as follows. When the sender inputs a message msg, the channel samples two uniform random bit strings msg_0 and msg_1 conditioned on $\mathsf{msg}_0 \oplus \mathsf{msg}_1 = \mathsf{msg}$; it delivers $(\mathsf{msg}_0, \mathsf{msg}_1)$ to the sender and the receiver (who recovers msg), and also delivers msg_0 to P_i and msg_1 to P_j. (This channel is implemented by the sender and the receiver, with the help of P_i and P_j.)

For the second step, each party $\widetilde{P}_{\{i,j\}}$ in $\widehat{\Pi}$ is replaced by a *secure implementation* via a Basic-GMW protocol execution between P_i and P_j using the setup variables (S_{ij}, S_{ji}). For this first P_i and P_j securely sample OT instances from the non-trivial setup (S_{ij}, S_{ji}); that this is possible follows from a result in [3], which showed that every non-trivial setup can be used to obtain oblivious transfer. Then P_i and P_j use the Basic-GMW protocol for securely realizing a *reactive functionality*. In a reactive functionality, (see the motivation of the security definition above), the trusted third party in the ideal scenario can interact with the two parties over multiple rounds, at each round sending them outputs sampled conditioned on the inputs and outputs in all previous rounds. ■

Remark: A special case of Theorem 1 appears in [4], where it was shown that, in a network with three parties, it is impossible to securely sample an OT pair between P_1 and P_2, if the setup consists only of OT pairs between P_1 and P_3 and between P_2 and P_3. The argument there could be reformulated as follows: by Theorem 1, it is enough to show that the flattened problem with parties $P_1, P_2, P_3, \widetilde{P}_{\{1,3\}}, \widetilde{P}_{\{2,3\}}$ (without any setup) cannot be securely realized; for this consider the partition of these 5 parties into two sets – say, $\{P_1, P_3, \widetilde{P}_{\{1,3\}}\}$ (subsumed by $\widetilde{P}_{\{1,3\}}$) and $\{P_2, \widetilde{P}_{\{2,3\}}\}$ (subsumed by $\widetilde{P}_{\{2,3\}}$) – so that the resulting sampling problem corresponds to samping an OT pair (without any setup), which is impossible.

V. On Necessary & sufficient conditions for secure sampling without setup

In the model with point-to-point communication channels, it remains open to characterize which distributions can be securely sampled without setups. One approach to ruling out the realizability of a distribution $(Y_1, \ldots, Y_m)$ is to consider partitioning $[m]$ in to $S \subseteq [m]$ and $\overline{S} = [m] \backslash S$, and rule out the realizability of the distribution $(Y_S, Y_{\overline{S}})$. However, we conjecture that in general, this is not enough to characterize secure realizability, for $m > 3$.

Conjecture 1. *For $m > 3$, there exists $p_{\mathbf{Y}} \in \mathfrak{P}^m$ such that for every subset $S \subseteq [m]$, the pair of random variables $(Y_S, Y_{\overline{S}})$ is securely realizable, but $\mathrm{Sam}_{\mathrm{pvt}}(p_{\mathbf{Y}})$ is not securely realizable.*

Nevertheless, for the case of $m = 3$, we show that the partitioning argument does give a full characterization.

Theorem 2. *For $p_{\mathbf{Y}} \in \mathfrak{P}^3$, $\mathrm{Sam}_{\mathrm{pvt}}(p_{\mathbf{Y}})$ is securely realizable if and only if for every subset $S \subseteq [3]$, the pair of random variables $(Y_S, Y_{\overline{S}})$ is securely realizable.*

The interesting direction in proving this is to show that if for all $S \subseteq [3]$, the pair of random variables $(Y_S, Y_{\overline{S}})$ is trivial (and hence pairwise securely realizable), then $\mathrm{Sam}_{\mathrm{pvt}}(p_{\mathbf{Y}})$ is securely realizable.

Since $Y_i, Y_{\overline{i}}$ is trivial, there exist random variables Q_i, defined by $p_{Q_i|Y_1,Y_2,Y_3}$, $i \in [3]$, such that $H(Q_i|Y_i) = H(Q_i|Y_{\overline{i}}) = 0$, and $I(Y_i; Y_{\overline{i}}|Q_i) = 0$. In the following simple protocol, the parties securely realize (Q_1, Q_2, Q_3) with a joint distribution induced by

$$p_{Y_1,Y_2,Y_3} \prod_{i=1}^{3} p_{Q_i|Y_1,Y_2,Y_3}, \tag{3}$$

and then each party locally samples Y_i given Q_i.

First, party P_1 locally samples Q_1 and sends msg_2 to P_2 and msg_3 to P_3 where $\mathsf{msg}_2, \mathsf{msg}_3$ are random strings uniformly distributed conditioned on $\mathsf{msg}_2 \oplus \mathsf{msg}_3 = \mathsf{bin}\,(Q_1)$, where $\mathsf{bin}\,(Q_1)$ is a binary string representation of Q_1. Also P_1 samples (several independent copies of) OT pairs $S_{2,3}, S_{3,2}$ and delivers $S_{2,3}$ to P_2 and $S_{3,2}$ to P_3. Now, P_2 and P_3 will use $(S_{2,3}, S_{3,2})$ as a setup in an execution of the Basic-GMW protocol to securely sample (Q_2, Q_3), conditioned on $\mathsf{bin}\,(Q_1) = \mathsf{msg}_2 \oplus \mathsf{msg}_3$ (where $\mathsf{msg}_2, \mathsf{msg}_3$ are their inputs in the Basic-GMW protocol). Finally, each party P_i samples Y_i conditioned on Q_i (using $p_{Y_i|Q_i}$ induced by (3)). It is easy to verify (using the that the resulting

The security of the protocol follows from a case analysis: correctness (i.e., the security condition for the set of corrupt parties $C = \emptyset$) easily follows from the correctness of the protocol used to sample (Q_1, Q_2, Q_3) and the fact that $I(Y_i; Y_{\overline{i}}|Q_i) = 0$. If only one party P_i is corrupt, its view in the protocol can be simulated given just Q_i (for $i = 2, 3$ this follows from the security of the Basic-GMW protocol), which

in turn can be determined from Y_i since $H(Q_i|Y_i) = 0$. If two parties, say $P_{\bar{i}}$ are corrupt, then in fact, they learn not only $Q_{\bar{i}}$ but also Q_i; however, since $H(Q_i|Y_{\bar{i}}) = 0$ as well, from $Y_{\bar{i}}$ all of Q_1, Q_2, Q_3 can be determined and then the view of the corrupt players can be perfectly simulated.

VI. Sampling in the Public Discussion Model

In this section, instead of the noiseless pairwise (point-to-point) private links in the previous sections, we consider a noiseless *public discussion channel*. In each round, each party sends a message over this channel which is received by all the other parties. In all other respects, the problem is the same as in Section II. We denote the problem (without setups) by $\mathrm{Sam}_{\mathrm{pub}}(p_{\mathbf{Y}})$. We say that a protocol Π *ϵ-securely realizes* $\mathrm{Sam}^t_{\mathrm{pub}}(p_{\mathbf{Y}})$, where $t \geq 1$, if for each $C \leq [m]$ with $|C| \leq t$, there exists a simulated view as in Definition 1 which satisfies (1)-(2). And, we say that $\mathrm{Sam}^t_{\mathrm{pub}}(p_{\mathbf{Y}})$ is *securely realizable* if for each $\epsilon > 0$, there is a protocol which ϵ-securely realizes $\mathrm{Sam}^t_{\mathrm{pub}}(p_{\mathbf{Y}})$.

Below, we give a characterization of the set of all distributions $p_{\mathbf{Y}}$ such that $\mathrm{Sam}^t_{\mathrm{pub}}(p_{\mathbf{Y}})$ is securely realizable, i.e., which can be sampled without a setup using public discussion. This is, in fact, a consequence of a more general result (presented in the full version), which includes as a special case the impossibility results in [15].

Theorem 3. *The following two conditions are equivalent:*

(a) $\mathrm{Sam}^t_{\mathrm{pub}}(p_{\mathbf{Y}})$ *is securely realizable (threshold $t \geq 1$ is arbitrary).*

(b) There exists a random variable Q jointly distributed with $Y_1, \ldots, Y_m$ (which have distribution $p_{\mathbf{Y}}$) such that the following conditions hold for all $i \in [m]$.

$$I(Y_i; Y_{\bar{i}}|Q) = 0, \tag{4}$$

$$I(Y_{\bar{i}}; Q|Y_i) = 0, \tag{5}$$

where $\bar{i} = [m] - \{i\}$ is the complement of $\{i\}$ in $[m]$.

Proof: (b)⇒(a): Consider the protocol where one of the parties samples Q privately and shares it with all the other parties (over the public discussion channel). Then, by (4), each party-i can sample their outputs using their private randomness using $p_{Y_i|Q}$ conditioned on the Q they received. Further, (5) implies that this protocol is also secure (for every threshold t).

(a)⇒(b): Our proof uses a generalization of the *monotone region* from [1]. For a set of m random variables $\mathbf{Z} = (Z_i)_{i\in[m]}$, we define a multi-dimensional region $\mathcal{R}(\mathbf{Z}) \subseteq \mathbb{R}^d_+$ (for an appropriate d) such that if $\{Z_i : i \in [m]\}$ are mutually independent random variables then $\mathcal{R}(\mathbf{Z}) = 0$, and for any protocol which (perfectly) securely realizes $\mathrm{Sam}^1_{\mathrm{pub}}(p_{\mathbf{Y}})$ (note that we set $t = 1$ which is the weakest form of (a)),

$$\mathcal{R}(\mathbf{V}_0) \subseteq \mathcal{R}(\mathbf{Y}), \tag{6}$$

where $\mathbf{V}_0$ is the initial view (consisting of independent private random variables at the parties) and $\mathbf{Y}$ is the output of the parties at the end of the protocol. Then it follows that the origin belongs to $\mathcal{R}(\mathbf{Y})$ as well. Further, this extends to protocols which securely realize $\mathrm{Sam}^1_{\mathrm{pub}}(p_{\mathbf{Y}})$ with vanishing error (rather than perfect security), if $\mathcal{R}(\mathbf{Z})$ is continuous in $p_{\mathbf{Z}}$.[2]

In the full version we show that the following $2m$-dimensional region satisfies the above requirements.

$$\begin{aligned}\mathcal{R}(\mathbf{Z}) = \{(r_{i1}, r_{i2})_{i\in[m]} : r_{i1} &\geq I(Z_i; Z_{\bar{i}}|Q_Z),\\ r_{i2} &\geq I(Z_{\bar{i}}; Q_Z|Z_i), \text{ for some } p_{Q_Z|Z_1,\ldots,Z_m}\}.\end{aligned}$$

The origin being in $\mathcal{R}(\mathbf{Y})$ is exactly condition (b).

Showing that the above definition of $\mathcal{R}$ satisfies (6) has two steps. The first step is to show that if $\mathbf{V}_{r-1} = (V_{r-1,i})_{i\in[m]}$ is the vector of views at the end of round-$r-1$ and $\mathbf{V}_r$ that at the end of round-r, then $\mathcal{R}(\mathbf{V}_{r-1}) \subseteq \mathcal{R}(\mathbf{V}_r)$. The second step is to show that $\mathcal{R}(\mathbf{V}) \subseteq \mathcal{R}(\mathbf{Y})$, where $\mathbf{V}$ is the final view of the parties in the protocol, and $\mathbf{Y}$ is "securely derived" from $\mathbf{V}$. The details are similar to the argument in [1] and are omitted for lack of space. ∎

References

[1] V. Prabhakaran and M. Prabhakaran, "Assisted common information with an application to secure two-party sampling," ArXiv, abs/1206.1282, 2012, preliminary versions appeared at ISIT 2010 and ISIT 2011. [Online]. Available: http://arxiv.org/abs/1206.1282

[2] O. Goldreich, *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[3] J. Kilian, "More general completeness theorems for secure two-party computation," in *STOC*, F. F. Yao and E. M. Luks, Eds. ACM, 2000, pp. 316–324.

[4] D. Harnik, Y. Ishai, and E. Kushilevitz, "How many oblivious transfers are needed for secure multiparty computation?" in *CRYPTO*, 2007, pp. 284–302.

[5] M. Fitzi, J. A. Garay, U. M. Maurer, and R. Ostrovsky, "Minimal complete primitives for secure multi-party computation," *J. Cryptology*, vol. 18, no. 1, pp. 37–61, 2005.

[6] E. Kushilevitz, "Privacy and communication complexity," *SIAM J. Discrete Math.*, vol. 5, no. 2, pp. 273–284, 1992.

[7] R. Künzler, J. Müller-Quade, and D. Raub, "Secure computability of functions in the IT setting with dishonest majority and applications to long-term security," in *TCC*, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 238–255.

[8] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Complexity of multi-party computation problems: The case of 2-party symmetric secure function evaluation," in *TCC*, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 256–273.

[9] B. Chor and Y. Ishai, "On privacy and partition arguments," *Inf. Comput.*, vol. 167, no. 1, pp. 2–9, 2001.

[10] Y. Wang and P. Ishwar, "On unconditionally secure multi-party sampling from scratch," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2011, pp. 1782–1786.

[11] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *STOC*, 1987, pp. 218–229, see [2, Chap. 7] for more details.

[12] J. Kilian, "Founding cryptography on oblivious transfer," in *STOC*, 1988, pp. 20–31.

[13] S. Haber and S. Micali, "Unpublished manuscript," 1986.

[14] O. Goldreich and R. Vainish, "How to solve any protocol problem - an efficiency improvement," in *CRYPTO*, 1987, pp. 73–86.

[15] A. A. Gohari and V. Anantharam, "Information-theoretic key agreement of multiple terminals: part i," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3973–3996, 2010.

[16] O. Reingold, Ed., *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, ser. Lecture Notes in Computer Science, vol. 5444. Springer, 2009.

[2] This is in fact much more general than is needed here and can be used to derive impossibility results for secure sampling with public discussion even when there is a non-trivial setup.