Indian Institute of Technology, Bombay

Communication and Randomness Lower Bounds for Secure Multiparty Computation

Srivatsan Sridhar
Roll No. 150070005

B.Tech. Project Report
under the guidance of

Prof. Sibi Raj Pillai
(Electrical Engineering, IIT Bombay)
Prof. Manoj Prabhakaran
(Computer Science and Engineering, IIT Bombay)
Prof. Vinod Prabhakaran
(School of Technology and Computer Science, TIFR)

May 2019

Abstract

Three-party secure computation is a problem wherein two mutually distrusting parties, Alice and Bob, wish to "securely" compute a function of their private data, with the help of a third party Charlie. Here, "securely" means that Alice and Bob do not learn the private data of each other, and Charlie does not learn their private data, apart from the function value. Secure multiparty computation (as considered in this project) is a generalization of this setting where we have k mutually distrusting users in place of Alice and Bob. We are interested in finding the minimum number of bits of messages exchanged between any two users (communication) and the minimum number of bits of randomness used for a secure protocol.

For three-party secure computation, Feige, Kilian and Naor (FKN) gave a general protocol for securely computing any function computable in non-deterministic logspace. These protocols extend to multiparty for some functions. However, it is known that these protocols are not optimal in terms of the amount of communication and randomness required, in general. Several information-theoretic methods have been used to derive lower bounds on the communication and randomness required. These bounds are also not always tight for general functions.

In this project, we consider three-party secure computation of a specific function, that is AND of two bits. It is observed that the randomness lower bound derived for this function is lower than the randomness used by the FKN protocol. However, we prove that in a non-interactive setting (where all parties send a single round of messages), the FKN protocol is optimal in the cardinality of the set of messages and randomness elements. In particular, we show that private randomization by the parties does not decrease the amount of common randomness used. We will also derive a bound on the entropy of the common randomness when private randomness is allowed. Further, we will relate this problem with other problems such as distribution design, and conclude by providing possible directions in which these bounds can be extended.

Contents

Chapter 1

Introduction

1.1 Secure Computation - Introduction

Information security is a major concern in today's world of communication. One important security problem, that is encrypting data to prevent eavesdroppers from gathering any information about the data, has been well studied. Such a problem is encountered in any public communication channel, and many security protocols have been developed, which are either perfectly secure or more often, computationally secure.

Here, we consider a different problem, where we require a third party to compute a function of our private data. That is, we want the third party (whom we do not trust) to receive enough information to be able to compute the required function, but to not learn anything more about the private data. This could arise because there are multiple parties, who do not individually have all the data required to compute the function, and do not trust each other (most common scenario). This could also happen in the case where the party with the private data is not computationally powerful to compute the function. The latter may be relevant for security constraints in machine learning based applications for smartphones where the computation is done on the server. More applications of this problem will be mentioned in Section 1.5.

1.2 Three-Party Secure Computation - Problem Definition

Referring to Figure 1.1, the three-party secure computation model can be described as follows. Alice (A) and Bob (B) respectively have private data (inputs) X and Y, drawn from finite alphabets $\mathcal{X}$ and $\mathcal{Y}$ according to a joint distribution $p_{XY}(x, y)$. Charlie (C) must compute $Z = f(x, y)$ where $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ is a function which takes values in the

finite alphabet $\mathcal{Z}$. We consider here that the function to be computed is a deterministic function of x and y.

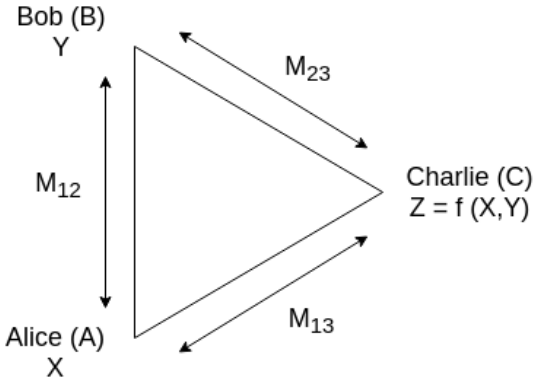

FIGURE 1.1: Description of the three-party secure computation model

Each pair of users is connected by a communication link, that is private from the other user. The messages exchanged between each pair of users are denoted as M_{ij} which belong to finite alphabets $\mathcal{M}_{ij}$ for $i, j = 1, 2, 3$. Here M_{ij} contains both messages sent from i to j and from j to i. A general secure computation protocol can run over multiple rounds, where the message sent by each user in a round has a distribution conditioned on its inputs (X or Y) and the messages it has seen so far. In the final round, Charlie outputs Z. The protocol is designed so as to terminate with probability 1. Here, each user can also sample its own private randomness, and hence the messages are specified by a distribution, and not a function. Such protocols that run over multiple rounds may be called interactive protocols. In this project however, only non-interactive or "one-shot" protocols are considered. In a general one-shot protocol, Alice first sends the message M_{12} to Bob. Then Alice and Bob send messages M_{13} and M_{23} respectively to Charlie. Charlie computes Z using M_{13} and M_{23}.

A valid protocol for three-party secure computation must satisfy the following correctness and secrecy conditions :

- **Correctness:** Charlie should compute the correct value of $Z = f(X, Y)$ where X and Y are the values of Alice's and Bob's inputs.
- **Secrecy against Alice:** Alice should not learn anything more about Y and Z than what is revealed by X, i.e. (Y, Z) are independent of (M_{12}, M_{13}) given X.
- **Secrecy against Bob:** Bob should not learn anything more about X and Z than what is revealed by Y, i.e. (X, Z) are independent of (M_{12}, M_{23}) given Y.
- **Secrecy against Charlie:** Charlie should not learn anything more about X and Y than what is revealed by Z, i.e. (X, Y) are independent of (M_{13}, M_{23}) given Z.

This model of security is known as the "honest-but-curious" model, where each party follows the protocol honestly, but may be curious to know the private data of other parties. In the above conditions, we assume that we require perfectly correct and perfectly secure computation, as against asymptotically correct and secure computation.

Our aim is to find the minimum amount of communication and randomness required for a perfectly secure protocol. The amount of communication is specified using the entropy of the messages $H(M_{ij})$ or using the cardinality of the message alphabets $|\mathcal{M}_{ij}|$ where we know that $\log_2 |\mathcal{M}_{ij}| \geq H(M_{ij})$. The amount of randomness can be defined as the additional entropy resulting from the protocol, given the inputs, i.e. $H(M_{12}, M_{13}, M_{23}, Z|X, Y)$.

1.3 The FKN Protocol

Feige, Kilian and Naor in [1], proposed a general protocol for secure computation of any function that is computable in nondeterministic logspace. This protocol is an example of a non-interactive or one-shot protocol. This protocol is also known as a private simultaneous messages (PSM) protocol. In the notation of the problem described above, this protocol can be given by the following steps:

Protocol 1.1. *FKN Protocol for a general function*

1. *Alice chooses $M_{12} \in \mathcal{M}_{12}$ according to a distribution $p_{M_{12}}(m_{12})$ and sends it to Bob privately. M_{12} is not revealed to Charlie.*
2. *Alice sends $M_{13} = m_{13}(X, M_{12})$, a deterministic function of X and M_{12}, to Charlie. M_{13} is not revealed to Bob.*
3. *Bob sends $M_{23} = m_{23}(Y, M_{12})$, a deterministic function of Y and M_{12}, to Charlie. M_{23} is not revealed to Alice.*
4. *Charlie computes $Z = f(X, Y)$ using M_{13} and M_{23}, as $Z = \widehat{f}(M_{13}, M_{23})$.*

This protocol satisfies the conditions for a perfectly secure and correct protocol if the following conditions are met :

- **Correctness:** $\widehat{f}(m_{13}(x, m_{12}), m_{23}(y, m_{12})) = f(x, y)\ \forall x \in \mathcal{X}, y \in \mathcal{Y}, m_{12} \in \mathcal{M}_{12}$. This, in turn, is possible iff $\text{supp}(M_{13}, M_{23}|X = x, Y = y) \cap \text{supp}(M_{13}, M_{23}|X = x', Y = y') = \Phi$ whenever $f(x, y) \neq f(x', y')$. ($\text{supp}(U)$ denotes the set of values of the random variable U with non-zero probability)

- **Secrecy against Alice:** satisfied since Alice does not receive any message.
- **Secrecy against Bob:** satisfied since M_{12} is chosen independent of X.
- **Secrecy against Charlie:** for all pairs of inputs (x_1, y_1) and (x_2, y_2) such that $f(x_1, y_1) = f(x_2, y_2)$, the distributions of the messages $(m_{13}(x_1, M), m_{23}(y_1, M))$ and $(m_{13}(x_2, M), m_{23}(y_2, M))$ are identical, where M is chosen randomly from $\mathcal{M}_{12}$.

In this protocol, since M_{12} is independent of both X and Y, and is available to both parties, we call it the common randomness. Both parties do not use any private randomness since M_{13} and M_{23} are given by deterministic functions. Thus the amount of randomness for such a protocol is given by $H(M_{12}, M_{13}, M_{23}, Z|X, Y) = H(M_{12})$ or it can be given by the cardinality $|\mathcal{M}_{12}|$. Note that this protocol can be equivalently described by saying that an external "dealer" picks the common randomness M_{12} independent of the inputs X and Y, and gives it to Alice and Bob. This can be used to easily extend such a protocol to the multiparty setting.

1.4 Multiparty Secure Computation - Extending the FKN Protocol

In general for multiparty computation, all parties can communicate with each other, and security attacks may involve collusion of several parties to find the private data of other parties, using the messages they have seen and their own private data. In such a scenario, it is known that perfectly secure computation is possible when less than half of the parties collude, if there is a private communication link between any pair of parties, and all parties have access to private randomness. In this project, such general protocols are not considered, but a simple extension to the FKN Protocol is considered.

We consider the multiparty setting where there are k parties denoted by A_i, each having their private data $X_i \in \mathcal{X}_i$ for $i = 1, ..., k$. Charlie (C) must compute a function $Z = f(X_1, ..., X_k)$. The FKN protocol can be extended in this manner, as shown in figure 1.2 :

Protocol 1.2. *FKN protocol for multiparty secure computation*

1. *A shared randomness M_0 drawn from a set $\mathcal{M}_0$ according to a distribution $p_{M_0}(m_0)$, independent of $X_1, \ldots, X_k$, is given to all parties A_i. M_0 is not revealed to Charlie. In this protocol, the amount of randomness is given by $H(M_0)$*

2. *Each party A_i sends $M_i = m_i(X_i, M_0)$, a deterministic function of its input and the shared randomness, to Charlie. M_i is not revealed to parties A_j with $j \neq i$.*

3. *Charlie computes $Z = f(X_1, ..., X_k)$ as $Z = \widehat{f}(M_1, ..., M_k)$*

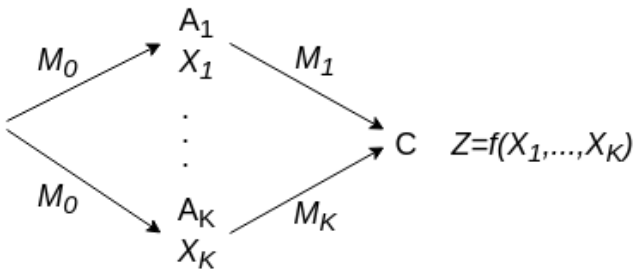

FIGURE 1.2: Description of the FKN protocol for multiparty secure computation

1.5 Applications of Secure Multiparty Computation

The problem of secure multiparty computation has interesting applications, some of which are discussed in [2]:

1. **Secure Audio Teleconferencing:** This was one of the earliest applications of secure MPC, studied in [3] and an improved solution proposed in [1]. In secure audio teleconferencing, three or more parties talk together on a telephone line. A bridge is required to facilitate this, whose role is to receive speech signals from all the parties, select the active signals (those that contain speech), and send to all parties the sum of the active signals. Alternatively, the bridge may select only the signal with maximum amplitude and send it to all parties. When the conversation needs to be encrypted, the bridge must be able to decrypt and encrypt the signals to perform its function, thus learning the signal itself. This is a problem if the bridge is not a trusted entity. Solutions proposed to this problem in [3] and [1] have the bridge securely compute either the sum or the max of the signals, without learning anything else about the signals.

2. **Secure Auctions:** In an auction, each buying party bids an amount for a certain product, while the selling party computes the maximum of these bids to decide who wins the auction. Each party has a maximal amount that they would pay, such that they stop bidding when the bid goes higher than that. This is their input. The parties run an interactive auction protocol so that the seller can find out the party with the highest maximal amount. For an honest auction, we require that this maximal amount be hidden from the seller and the other buyers. Otherwise the seller could force the winning party to pay an amount as high as his maximal

amount, or another buyer could win the auction by paying an amount barely higher than the other parties' maximal amount. Thus this is an example of secure multiparty computation.

3. **Benchmark Analysis:** Several companies may be interested to have a third party evaluate their performance with respect to their competitors. Each company has certain parameters such as their profits, productivity, salaries etc. as inputs and wish to keep these private from the other companies and the third party. This is also a scenario for secure MPC.

4. **Machine Learning:** A more recent application is where an organization uses data such as content in emails, movies watched, usage patterns, images etc. to train machine learning models for applications such as spam filters, movie recommendation and image recognition systems. Such data is collected from a large number of users. Recently, people are becoming concerned about the security threat of allowing these organizations to access one's personal information. The user wants his/her data to be private, while the organization wants to train accurately on the data. Here, the function to be computed could be a complex feature extraction from the data.

Chapter 2

Literature Study

2.1 Existence of Secure Computation Protocols

Three-party Secure Computation was studied by Feige, Kilian and Naor in [1]. The main results of this work were towards showing that any function can be computed securely in the model described in that work. Further they showed that for functions computable in nondeterministic logspace, there exist efficient secure computation protocols.

The first result from [1] shows that any function $f(x, y)$ can be computed securely. However, the amount of communication and randomness may be exponential. The result is shown for functions of the form $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$. However it is still general because any other finite input and output alphabets can be coded as binary vectors, and each bit of the output can be computed as a separate binary output function.

This result is shown by constructing a protocol. The bipartite graph representing f is considered, where the vertices in one partition correspond to values of X in $\{0,1\}^n$ and the vertices in the other partition correspond to values of Y in $\{0,1\}^n$. There is an edge between the values x and y iff $f(x, y) = 1$. The protocol draws a common random string of $n + 2^n$ bits, and transforms this graph based on the common string. 2^n bits of the common randomness are called r_y associated with each vertex y. All edges incident on vertex y are complemented if $r_y = 1$. The remaining n bits are denoted by π and used to define a cyclic permutation of the vertices corresponding to Y, i.e. y is permuted to $y - \pi$. Alice and Bob send the following messages to Charlie:

$$M_{13} = m_{13}(y, \pi, r) = (\, f(a, \pi) \oplus r_\pi, f(a, \pi + 1) \oplus r_{\pi+1}, \ \ldots \ , f(a, \pi - 1) \oplus r_{\pi-1} \,) \quad (2.1)$$

$$M_{23} = m_{23}(y, \pi, r) = (\, y - \pi \bmod 2^n, r_y \,) \quad (2.2)$$

Charlie computes Z by choosing the $(y - \pi \bmod 2^n)^{\text{th}}$ entry in M_{13} and complementing it iff r_y is 1.

This protocol can be verified to be correct and secure with reference to the conditions given in Section 1.3. The amount of communication used is 2^n bits for M_{13} and $n+1$ bits for M_{23}. The amount of randomness used ($H(M_{12})$) is $2^n + n$ bits. Thus, these may be exponential in the input length n, in general.

The second result shows that for any function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that can be computed in nondeterministic logspace, there exists a secure computation protocol in which the amount of communication and randomness, and all the computations are polynomial in n. A decision problem is said to be computable in nondeterministic logspace if it can be computed by a nondeterministic Turing machine using an amount of memory that is logarithmic in n. Again, a general protocol is constructed in [1].

2.2 FKN Protocol for Secure Computation of AND

Another important result from [1] is the construction of a non-interactive multiparty secure computation protocol for the logical AND of k bits. There are k parties, each having an input $X_i \in \{0,1\}$. Charlie has to compute $Z = X_1 \wedge X_2 \wedge \ldots \wedge X_k$. The protocol is described below, using the notation defined in Section 1.4

Protocol 2.1. *FKN protocol for multiparty secure computation of AND*

1. *The shared randomness M_0 consists of a prime $p > k$, a number r such that $0 < r < p$, and $r_1, \ldots, r_k$ such that $\sum_{i=1}^{k} r_i = 0 \bmod p$*
2. *Each party sends $M_i = r(1 - x_i) + r_i \bmod p$ if its input is x_i.*
3. *Charlie outputs $Z = 1$ iff $\sum_{i=1}^{k} M_i = 0 \bmod p$.*

The common randomness includes k numbers which sum to 0 mod p ($k-1$ of them can be chosen independently), and one non-zero number mod p. Thus the amount of common randomness is $(k-1)\log_2 p + k \log_2(p-1)$ bits. The communication by each party is $\log_2 p$ bits. Alternatively, we can look at the cardinality of the alphabets of the common randomness and messages, and thus $|\mathcal{M}_0| = p^{k-1}(p-1)^k$ and $|\mathcal{M}_i| = p$ for this protocol. By Betrand's postulate, there will exist a prime p such that $k < p < 2k$. Therefore the amount of randomness is $O(k \log k)$ bits, and the amount of communication on each link is $O(k)$ bits.

For the three-party setup (with only two users Alice and Charlie), this protocol can be rephrased in the following manner, using the notation in Section 1.3:

Protocol 2.2. *FKN protocol for three-party secure computation of AND*

1. *M_{12} is a randomly and uniformly picked permutation of (0,1,2) - say (α, β, γ). This is sent by Alice to Bob.*
2. *$M_{13} = \alpha$ if $X = 1$ and β if $X = 0$. Alice sends M_{13} to Charlie.*
3. *$M_{23} = \alpha$ if $Y = 1$ and γ if $Y = 0$. Bob sends M_{23} to Charlie.*
4. *Charlie computes $Z = 1$ if $M_{13} = M_{23}$, and $Z = 0$ otherwise.*

This particular protocol is the special case of $k = 2$ and $p = 3$ in the above general protocol. Thus the cardinality of the randomness set is $|\mathcal{M}_{12}| = 6$ ($\log_2 6$ bits) and the cardinality of the message alphabets are $|\mathcal{M}_{13}| = |\mathcal{M}_{23}| = 3$ ($\log_2 3$ bits).

2.3 Information Theoretic Bounds for General Functions

The authors in [4] have used information theoretic methods to obtain lower bounds on the communication and randomness for three-party secure computation of general functions. That work considers interactive (multi-round) protocols where each party has access to private randomness as well. General functions are considered over any finite input alphabets $\mathcal{X}$ and $\mathcal{Y}$. The function is allowed to be randomized, i.e. Z is specified by a probability distribution $p_{Z|XY}(z|x, y)$. Each user has a block of inputs $X^n \in \mathcal{X}^n$ and $Y^n \in \mathcal{Y}^n$. Thus the correctness condition here means that the output Z^n should be distributed according to $\prod_{i=1}^{n} p_{Z|XY}(z_i|x_i, y_i)$.

The main result of [4] uses the *residual information* between pairs of X, Y and Z to derive progressively tighter lower bounds under tighter assumptions on the function. The tightest set of bounds derived for the entropies of the messages $H(M_{12})$, $H(M_{13})$ and $H(M_{23})$ (which will be used for the AND function in the following sections of this thesis), is when the input distribution p_{XY} has full support and the *characteristic bipartite graph* of the distributions p_{XY}, p_{YZ} and p_{ZX} are connected. The AND function satisfies the latter condition if the former is true.

These lower bounds when evaluated for the AND function, where $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and $Z = X \wedge Y$, give

$$H(M_{13}) \geq \log_2 3 \qquad H(M_{23}) \geq \log_2 3 \qquad H(M_{12}) \geq 1.826 \tag{2.3}$$

While the lower bounds for $H(M_{13})$ and $H(M_{23})$ are achieved by the FKN protocol, the lower bound for $H(M_{12})$ is not achieved (the protocol uses $H(M_{12}) = \log_2 6 \approx 2.585$).

This leaves the question open, of whether this lower bound is tight for the AND function or not. This question is analyzed in [5] and in this project.

2.4 Randomness Lower Bound for Non-interactive Secure Computation of AND Without Private Randomness

The work in [5], in a way, extends from [4]. In particular, they prove that $\log_2 6$ is a tight lower bound for $H(M_{12})$ when the FKN protocol is used (non-interactive protocol where parties do not use private randomness). They start by showing that the cardinality $|M_{12}| \geq 6$ using the bounds derived for AND in [4], and a counting argument.

When Alice and Bob do not use private randomness, M_{13} and M_{23} are deterministic functions of M_{12}, X and Y : $M_{13} = m_{13}(M_{12}, X)$, $M_{23} = m_{23}(M_{12}, Y)$. The lower bound $H(M_{13}) \geq \log_2 3$ means that $|\text{supp}(M_{13})| \geq 3$, where $\text{supp}(X)$ denotes the support of the random variable X, i.e. $\{x \in \mathcal{X} : p_X(x) > 0\}$. The lower bound for $|\mathcal{M}_{12}|$ is derived using the following properties of a secure protocol, mentioned in Section 1.3:

1. $f(1,0) \neq f(1,1)$: For Charlie to compute Z with zero probability of error, we require $\text{supp}((M_{13}, M_{23})|XY = 10)$ and $\text{supp}((M_{13}, M_{23})|XY = 11)$ to be disjoint. (Correctness)
2. $f(0,0) = f(0,1)$: For Charlie to not distinguish between the inputs $(0,0)$ and $(1,0)$, we require $\text{supp}((M_{13}, M_{23})|XY = 00) = \text{supp}((M_{13}, M_{23})|XY = 01)$. (Secrecy)

Fix $M_{12} = m_{12}$ and let $a = m_{13}(m_{12}, 0)$, $b = m_{23}(m_{12}, 0)$ and $b' = m_{23}(m_{12}, 1)$.Then:

$$[a, b] \in \text{supp}((M_{13}, M_{23})|XY = 00)$$
$$[a, b'] \in \text{supp}((M_{13}, M_{23})|XY = 01)$$

From property 1, $b \neq b'$ and from property 2,

$$[a, b'] \in \text{supp}((M_{13}, M_{23})|XY = 00)$$

Since a appears as $[a, b]$ and $[a, b']$ in $\text{supp}((M_{13}, M_{23})|XY = 00)$, there exists $m'_{12} \in \mathcal{M}_{12}$ such that $a = m_{13}(m'_{12}, 0)$. Thus, for every element $a \in \text{supp}(M_{13}|X = 0)$, there are at least two elements in $\mathcal{M}_{12}$. Further from property 2, $\text{supp}(M_{13}|X = 0) = \text{supp}(M_{13}|X = 1) = \text{supp}(M_{13})$. This gives the lower bound

$$|\mathcal{M}_{12}| \geq 2|\text{supp}(M_{13})| \geq 6 \tag{2.4}$$

In a similar manner, the lower bound $H(M_{12}) \geq \log_2 6$ is shown for the same class of protocols. In particular, for a given $x \in \mathcal{X}$, let $S \subseteq \mathcal{Y}$ such that $f(x,y) = f(x,y')$ for all $y, y' \in S$. Then

$$H(M_{12}) \geq H(M_{13}|X = x) + \log_2 |S| \tag{2.5}$$

For AND, by choosing $x = 0$ and $S = \{0,1\}$, this evaluates to $H(M_{12}) \geq \log_2 6$. Thus, the FKN protocol for three-party secure computation of AND is optimal in both communication and randomness when we restrict to non-interactive protocols without private randomness. The authors in [5] left this question open, as to whether using private randomness could help reduce $H(M_{12})$ or $|\mathcal{M}_{12}|$ used in the protocol. This question will be answered in this project. In fact, they provide an example where this can be done. This example is discussed in the next section.

2.5 Example of a Protocol with Private Randomness

This function is an example where lesser $H(M_{12})$ can be used when the parties use private randomness, than when they do not. Consider the following function with $X, Y \sim \text{Unif}\{0,1,2\}$:

$$f(X,Y) = \begin{cases} 2 & \text{if } X = 2 \text{ or } Y = 2 \\ X \oplus Y & \text{otherwise} \end{cases}$$

Using the bounds from [4], it can be shown that $H(M_{13}) \geq 2.3137$ and $H(M_{23}) \geq 2.3137$. Further using the bound in 2.5, we get $H(M_{12}) \geq 3.8987$. However, there exists a protocol (shown below) given in [5] which uses private randomness by both Alice and Bob, and achieves $H(M_{13}) = H(M_{23}) = \log_2 3 + 1 \approx 2.5850$ and $H(M_{12}) = \log_2 6 + 1 \approx 3.5850$. Thus, using private randomness has the potential to lower the amount of communication on the 1-2 link. In the next chapter, we will show that for secure computation of AND, using private randomness cannot reduce $|\mathcal{M}_{12}|$ below 6.

Protocol 2.3. *Protocol using private randomness (k' and k'' are drawn from Alice's and Bob's private randomness)*

1. *M_{12} contains a uniformly chosen permutation of $(0,1,2)$, say (α, β, γ) and a uniform bit k.*
2. *Alice sends $M_{13} = (\alpha, X \oplus k)$ if $X \in \{0,1\}$, and (β, k') if $X = 2$.*
3. *Bob sends $M_{23} = (\alpha, Y \oplus k)$ if $Y \in \{0,1\}$, and (γ, k'') if $Y = 2$.*
4. *Charlie finds $Z = 2$ if $M_{13}(1) = M_{23}(1)$, and $Z = M_{13}(2) \oplus M_{23}(2)$ otherwise.*

Chapter 3

Randomness Lower Bounds Derived in This Project

In this chapter, we consider the three-party secure computation model for the AND function. We examine the lower bound $|\mathcal{M}_{12}| \geq 6$ that was shown in [4] and recapitulated in Section 2.4. That proof considered non-interactive protocols without private randomness. In this proof, we allow the parties to use private randomness and then examine if $|\mathcal{M}_{12}| \geq 6$ is necessary for a perfectly secure protocol.

3.1 Randomness Lower Bound When Only One Party Uses Private Randomness

Theorem 3.1. *When only one party uses private randomness,* $|\mathcal{M}_{12}| \geq 6$ *for a non-interactive three-party secure computation protocol for AND.*

Proof : Suppose only Bob is allowed private randomness, M_{13} is a deterministic function of M_{12} and X, while M_{23} is chosen from a probability distribution $p_{M_{23}|M_{12}Y}$. Thus for each value of M_{12} and Y, there is a distribution from which Bob chooses M_{23}.

Fix $M_{12} = m_{12}$ and let $a = m_{13}(m_{12}, 0)$. Let

$$b \in \text{supp}(M_{23}|Y = 0, M_{12} = m_{12})$$
$$b' \in \text{supp}(M_{23}|Y = 1, M_{12} = m_{12})$$

be representative elements of the distributions that they are chosen from. From property 1 in Section 2.4, $b \neq b'$, and from property 2,

$$[a, b'] \in \text{supp}((M_{13}, M_{23})|XY = 00)$$

The same argument used in Section 2.4 can be continued here. Since a appears as $[a, b]$ and $[a, b']$ in $\text{supp}((M_{13}, M_{23})|XY = 00)$, for every element a in $\text{supp}(M_{13}|X = 0)$, there exists $m_{12}, m'_{12} \in \mathcal{M}_{12}$ such that $a = m_{13}(m_{12}, 0) = m_{13}(m'_{12}, 0)$. Since $\text{supp}(M_{13}|X = 0) = \text{supp}(M_{13}|X = 1) = \text{supp}(M_{13})$, we get the same bound

$$|\mathcal{M}_{12}| \geq 2|\text{supp}(M_{13})| \geq 6 \tag{3.1}$$

Since Alice and Bob are equivalent in this protocol, the same can be said about private randomization by Alice. Thus private randomization by one party alone does not reduce the amount of shared randomness required.

3.2 Randomness Lower Bound When Both Parties Use Private Randomness

When both Alice and Bob can use private randomness, the above argument does not hold directly. However, a more general approach can be still used to show that a size of less than 6 for $\mathcal{M}_{12}$ cannot satisfy secrecy and correctness, by using a counting argument based on the support sets of M_{13} and M_{23} under various inputs.

Theorem 3.2. *When both parties are allowed to use private randomness,* $|\mathcal{M}_{12}| \geq 6$ *for a non-interactive three-party secure computation protocol for AND.*

Proof :

Setup of the proof

The two conditions for correctness and secrecy (given in Section 2.4) continue to hold as follows:

1. Correctness requires that

$$\text{supp}((M_{13}, M_{23})|XY = 10) \cap \text{supp}((M_{13}, M_{23})|XY = 11) = \Phi \tag{3.2}$$

In this case, we can equivalently say that

$$\text{supp}((M_{13}, M_{23})|XY = 10, M_{12} = m_1) \cap \text{supp}((M_{13}, M_{23})|XY = 11, M_{12} = m_2) = \Phi \tag{3.3}$$

for any $m_1, m_2 \in \mathcal{M}_{12}$.

2. Secrecy requires that

$$\text{supp}((M_{13}, M_{23})|XY = 00) = \text{supp}((M_{13}, M_{23})|XY = 01) = \text{supp}((M_{13}, M_{23})|XY = 10) \tag{3.4}$$

We will use the notation A_m^x to denote $\text{supp}(M_{13}|X = x, M_{12} = m)$ and B_m^y to denote $\text{supp}(M_{23}|Y = y, M_{12} = m)$ for $x, y \in \{0, 1\}$ and $m \in \mathcal{M}_{12}$. Thus $\text{supp}((M_{13}, M_{23})|XY = xy, M_{12} = m)) = A_m^x \times B_m^y$ which is denoted by $A_m^x B_m^y$.

The approach used will be to construct the protocol by listing $A_m^x B_m^y$ for each value of m, x and y while satisfying conditions 1 and 2 above. This is done in Table 3.1. Each row represents a combination of inputs X and Y and each column represents an element of $\mathcal{M}_{12}$. Elements of $\mathcal{M}_{12}$ are indexed as 1,2, and so on. Each cell shows the support set $\text{supp}((M_{13}, M_{23})|XY = xy, M_{12} = m)) = A_m^x B_m^y$.

Corresponding to the two conditions given above, we have to fulfill these two objectives while filling the table.

1. According to condition 1 above, each of the sets in the first 3 rows of the table are disjoint from each of the sets in the last row.
2. According to condition 2, all elements of the sets in row 1 must also be present in the sets in rows 2 and 3 (similarly for rows 2 and 3).

	$M_{12} = 1$	$M_{12} = 2$	$M_{12} = 3$	$M_{12} = 4$	$M_{12} = 5$	$M_{12} = 6$
$XY = 00$	$A_1^0 B_1^0$	$A_2^0 B_2^0$	$A_3^0 B_3^0$	$A_4^0 B_4^0$	$A_5^0 B_5^0$	$A_6^0 B_6^0$
$XY = 01$	$A_1^0 B_1^1$	$A_2^0 B_2^1$	$A_3^0 B_3^1$	$A_4^0 B_4^1$	$A_5^0 B_5^1$	$A_6^0 B_6^1$
$XY = 10$	$A_1^1 B_1^0$	$A_2^1 B_2^0$	$A_3^1 B_3^0$	$A_4^1 B_4^0$	$A_5^1 B_5^0$	$A_6^1 B_6^0$
$XY = 11$	$A_1^1 B_1^1$	$A_2^1 B_2^1$	$A_3^1 B_3^1$	$A_4^1 B_4^1$	$A_5^1 B_5^1$	$A_6^1 B_6^1$

TABLE 3.1: Table for construction of secure AND protocol

We will observe which sets of messages cannot occur in the same cell as they would violate objective 1 if they did so. We will fill elements in the table so as to satisfy

objective 2. In this way, we can demonstrate that any cardinality $|\mathcal{M}_{12}| < 6$ cannot satisfy objectives 1 and 2. To do this, we fill Table 3.1 by giving each support set a unique name. We will then impose the relations that these sets have among each other.

Assumptions

Without loss of generality we fill the first column of Table 3.1 with arbitrary $A_1^0, A_1^1, B_1^0, B_1^1$. Note that A_1^0 and A_1^1 are disjoint and B_1^0 and B_1^1 are disjoint. **To begin with, we at least need 1 bit of randomness to ensure secrecy, so there are at least 2 columns in the table.** If we wish to have $|\mathcal{M}_{12}| = 2$, then for $M_{12} = 2$, we need $A_1^0 B_1^1 \subseteq A_2^0 B_2^0$ and $A_1^1 B_1^0 \subseteq A_2^0 B_2^0$ to fulfill objective 2. Thus $A_1^0, A_1^1 \subseteq A_2^0$ and $B_1^0, B_1^1 \subseteq B_2^0$ This would mean that $A_1^1 B_1^1 \subseteq A_2^0 B_2^0$ which violates objective 1. Thus we need more than 2 elements in $\mathcal{M}_{12}$.

We will prove that a size of $|\mathcal{M}_{12}| \leq 5$ cannot satisfy objectives 1 and 2 together. Since the sets $A_1^0 B_1^1$ and $A_1^1 B_1^0$ must appear in Row 1, We begin by assuming without loss of generality that

$$A_1^1 \cap A_2^0 \neq \Phi \qquad B_1^0 \cap B_2^0 \neq \Phi \qquad A_1^0 \cap A_3^0 \neq \Phi \qquad B_1^1 \cap B_3^0 \neq \Phi \tag{3.5}$$

This further implies (to satisfy objective 1), that

$$B_1^1 \cap B_2^0 = \Phi \qquad B_1^1 \cap B_2^1 = \Phi \qquad A_1^1 \cap A_3^0 = \Phi \qquad A_1^1 \cap A_3^1 = \Phi \tag{3.6}$$

Constraints on splitting the sets among multiple columns

With respect to the next two columns (4 and 5), consider the following cases:

1. If $A_1^1 \cap A_4^0 = A_1^1 \cap A_5^0 = \Phi$, then we require that $A_1^1 \subseteq A_2^0$ and $B_1^0 \subseteq B_2^0$. Therefore $B_1^0 \cap B_2^1 = \Phi$. These conditions along with (3.6) will prevent $A_1^1 B_1^0$ from appearing in row 2. A similar contradiction can be shown when $B_1^1 \cap B_4^0 = B_1^1 \cap B_5^0 = \Phi$. **This case also shows us that $|\mathcal{M}_{12}| \leq 4$ cannot be possible**, because if we have less than 5 columns, we can consider the extra columns with empty sets.

2. If $A_1^1 \cap A_4^0 \neq \Phi, A_1^1 \cap A_5^0 \neq \Phi$, this implies $B_1^1 \cap B_4^0 = B_1^1 \cap B_5^0 = \Phi$ which is already contradicted in case 1. A similar contradiction can be shown when $B_1^1 \cap B_4^0 \neq \Phi, B_1^1 \cap B_5^0 \neq \Phi$.

3. Cases 1 and 2 tell us that exactly one of $A_1^1 \cap A_4^0$ and $A_1^1 \cap A_5^0$ is empty. Similarly, exactly one of $B_1^1 \cap B_4^0$ and $B_1^1 \cap B_5^0$ is empty. Without loss of generality again,

let us assume:

$$A_1^1 \cap A_4^0 \neq \Phi \qquad A_1^1 \cap A_5^0 = \Phi \qquad B_1^1 \cap B_4^0 = \Phi \qquad B_1^1 \cap B_5^0 \neq \Phi \tag{3.7}$$

From (3.5)-(3.7) and objective 2, we know that $A_1^1 B_1^0 \subseteq A_2^0 B_2^0 \cup A_4^0 B_4^0$. Thus the elements of $A_1^1 B_1^0$ may be split among $A_2^0 B_2^0$ and $A_4^0 B_4^0$. We will show that at most one of A_1^1 or B_1^0 can be split into two non-full subsets across the two columns. Let

$$A_1^1 \cap A_2^0 = \delta A_1^1 \qquad A_1^1 \cap A_4^0 = \delta' A_1^1 \qquad B_1^0 \cap B_2^0 = \delta B_1^0 \qquad B_1^0 \cap B_4^0 = \delta' B_1^0 \tag{3.8}$$

where δS and $\delta' S$ denote non-full subsets of the set S, such that $\delta S \cup \delta' S = S$. Then we can see that there would be elements in $\delta A_1^1 \delta' B_1^0$ and $\delta' A_1^1 \delta B_1^0$ which are not covered by $A_2^0 B_2^0 \cup A_4^0 B_4^0$. More generally, we may allow $\delta' A_1^1$ and δB_1^0 to be equal to their full sets, and yet there will exist elements in $(A_1^1 \setminus \delta A_1^1)(B_1^0 \setminus \delta' B_1^0)$ which won't be covered. The remaining possible cases for the distribution of elements of $A_1^1 B_1^0$ and $A_1^0 B_1^1$ in the first row are shown in Table 3.2, where δ and δ' have the same meaning as described above.

	(a)	(b)
(i)	$A_1^1 \cap A_2^0 = \delta A_1^1, A_1^1 \cap A_4^0 = \delta' A_1^1$	$B_1^1 \cap B_3^0 = \delta B_1^1, B_1^1 \cap B_5^0 = \delta' B_1^1$
(ii)	$B_1^0 \cap B_2^0 = \delta B_1^0, B_1^0 \cap B_4^0 = \delta' B_1^0$	$A_1^0 \cap A_3^0 = \delta A_1^0, A_1^0 \cap A_5^0 = \delta' A_1^0$
(iii)	$A_1^1 \subseteq A_2^0, B_1^0 \subseteq B_2^0$	$A_1^0 \subseteq A_3^0, B_1^1 \subseteq B_3^0$
(iv)	$A_1^1 \subseteq A_4^0, B_1^0 \subseteq B_4^0$	$A_1^0 \subseteq A_5^0, B_1^1 \subseteq B_5^0$

TABLE 3.2: Exhaustive cases for distribution of the elements of $A_1^1 B_1^0$ and $A_1^0 B_1^1$ in row 1

Exhaustive analysis of all possible assignments

Now, we shall prove that each of the cases in Table 3.2 will fail to satisfy objectives 1 and 2.

1. **Case (a)(i) :** To cover all elements of $A_1^0 B_1^1$ in row 1, we need

$$\delta A_1^1, \delta' A_1^1 \neq \Phi \qquad B_1^0 \subseteq B_2^0, B_1^0 \subseteq B_4^0$$
$$\implies B_1^0 \cap B_2^1 = B_1^0 \cap B_4^1 = \Phi$$

Using this result, (3.6) and (3.7), $A_1^1 B_1^0$ cannot appear in row 2. Case (b)(i) can be disproved in the same manner.

2. **Case (a)(ii) and (b)(ii) together :** To cover all elements of $A_1^0B_1^1$ and $A_1^1B_1^0$ in row 1, we need

$$\begin{aligned} \delta B_1^0, \delta' B_1^0 \neq \Phi \qquad & A_1^1 \subseteq A_2^0, A_1^1 \subseteq A_4^0 \\ \delta A_1^0, \delta' A_1^0 \neq \Phi \qquad & B_1^1 \subseteq B_3^0, B_1^1 \subseteq B_5^0 \\ \implies A_1^1 \cap A_2^1, A_3^0, A_3^1, A_4^1, A_5^0, A_5^1 = \Phi \qquad & B_1^1 \cap B_2^0, B_2^1, B_3^1, B_4^0, B_4^1, B_5^1 = \Phi \end{aligned}$$

where some of the disjointness conditions come from the assumptions in (3.6) and (3.7), while the others hold for this case. Using these restrictions, we observe that the only possible assignment to cover the following subsets is as follows:

$$\begin{aligned} A_1^1B_1^0 \text{ in row 2} &\implies \delta' B_1^0 \subseteq B_2^1, \delta B_1^0 \subseteq B_4^1 \\ A_1^0B_1^1 \text{ in row 3} &\implies \delta' A_1^0 \subseteq A_3^1, \delta A_1^0 \subseteq A_5^1 \\ A_1^0B_1^0 \text{ in row 2} &\implies A_1^0 \subseteq A_2^0, A_1^0 \subseteq A_4^0 \\ A_1^0B_1^0 \text{ in row 3} &\implies B_1^0 \subseteq B_3^0, B_1^0 \subseteq B_5^0 \\ &\implies A_2^1 \cap A_1^0, A_1^1 = \Phi \end{aligned}$$

The last condition comes due to objective 1, and means that A_2^1 contains fresh elements that have not been seen before in A_1^0 or A_1^1 (since A_2^1 must be non-empty). However $A_2^1\delta B_1^0$ cannot appear anywhere in row 1 since $A_2^1\delta' B_1^0$ must not appear in row 1.

3. **Case (a)(ii) and (b)(iii) together :** To cover all elements of $A_1^0B_1^1$ and $A_1^1B_1^0$ in row 1, we need

$$\begin{aligned} \delta B_1^0, \delta' B_1^0 \neq \Phi \qquad & A_1^1 \subseteq A_2^0, A_1^1 \subseteq A_4^0 \qquad A_1^0 \subseteq A_3^0, B_1^1 \subseteq B_3^0 \\ \implies A_1^1 \cap A_2^1, A_3^0, A_3^1, A_4^1, A_5^0, A_5^1 = \Phi \qquad & B_1^1 \cap B_2^0, B_2^1, B_3^1, B_4^0, B_4^1 = \Phi \end{aligned}$$

From the assumptions in (3.5)-(3.7) and the above conditions, we have the following assignments:

$$\begin{aligned} A_1^1B_1^0 \text{ in row 2} &\implies \delta' B_1^0 \subseteq B_2^1, \delta B_1^0 \subseteq B_4^1 \\ &\implies A_2^1 \cap A_1^0 = A_2^1 \cap A_1^1 = \Phi \\ A_1^0B_1^1 \text{ in row 3} &\implies A_1^0 \subseteq A_5^1, B_1^1 \subseteq B_5^0 \\ A_1^0B_1^0 \text{ in row 3} &\implies B_1^0 \subseteq B_5^0 \\ A_2^1\delta B_1^0 \text{ in row 1} &\implies A_2^1 \subseteq A_3^0, \delta B_1^0 \subseteq B_3^0 \\ A_2^1\delta B_1^0 \text{ in row 2} &\implies A_2^1 \subseteq A_3^0, \delta B_1^0 \subseteq B_3^1 \end{aligned}$$

However the last assignment is a contradiction since we require $B_3^0 \cap B_3^1$. We can use a similar proof when case (iii) is replaced by (iv) or when (a) and (b) are interchanged.

4. **Case (a)(iii) and (b)(iii) together :** From the assumptions (3.5)-(3.7) and the assumptions for this case, we get:

$$B_2^1 \cap B_1^0 = \Phi, B_2^1 \cap B_1^1 = \Phi$$
$$A_3^1 \cap A_1^0 = \Phi, A_3^1 \cap A_1^1 = \Phi$$
$$A_2 B_1^0 \text{ in row } 2 \implies A_1^1 \subseteq A_4^0, B_1^0 \subseteq B_4^1$$
$$A_6 B_1^1 \text{ in row } 2 \implies A_3^1 \subseteq A_5^0, B_1^1 \subseteq B_5^1$$

However the last assignment is a contradiction since we require $B_5^0 \cap B_5^1$. We can use a similar proof when case (iii) is replaced by (iv) and when (a) and (b) are interchanged.

One can verify now that all possible assignments of the elements of $A_1^0 B_1^1$ and $A_1^1 B_1^0$ in the first row have been contradicted above. (This can be done easily by seeing that each of the δ and δ' subsets defined above can be the full set or not the full set, and the above considered cases account for all 16 of those possibilities). Thus, we have shown through a counting argument that a common randomness of size $|\mathcal{M}_{12}| \leq 5$ does not suffice for this problem. Since we know that a protocol exits with $|\mathcal{M}_{12}| = 6$, we can give the tight lower bound $|\mathcal{M}_{12}| \geq 6$.

3.3 Entropy Lower Bound When Both Parties Use Private Randomness

In the previous section, we established the cardinality bound $|\mathcal{M}_{12}| \geq 6$ for a perfectly secure three-party secure computation protocol with private randomness for both users. This does not directly establish the entropy bound $H(M_{12}) \geq \log_2 6$ under the same circumstances.

In a manner similar to how (2.5) was shown in [5], the following entropy bound is developed. We consider secure computation of an arbitrary function $f(x, y)$ with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

Theorem 3.3. *Let $x \in \mathcal{X}$ and $S_x \subseteq \mathcal{Y}$ such that $f(x, y) = f(x, y')$ for all $y, y' \in S_x$. Then*

$$H(M_{12}) \geq I(M_{12}; M_{13}|X = x) + \log_2 |S_x| \tag{3.9}$$

Proof : Let $x \in \mathcal{X}$, $S_x = \{y_1, \dots y_{|S_x|}\}$, and $m_{12} \in \mathcal{M}_{12}$ such that $[a, b_i] \in \text{supp}((M_{13}, M_{23})|M_{12} = m_{12}, X = x, Y = y_i)$. Because we have private randomness and a non-interactive protocol, M_{13} or M_{23} are given by distributions

$$Pr(m_{13}, m_{23}|m_{12}, x, y) = Pr(m_{13}|m_{12}, x)Pr(m_{23}|m_{12}, y) \tag{3.10}$$

Because of the condition for security, the distribution of the messages should be identical for all (x, y_i) such that $y_i \in S$. Thus,

$$Pr(M_{13} = a, M_{23} = b_i|X = x, Y = y_i) = Pr(M_{13} = a, M_{23} = b_i|X = x, Y = y_1) \tag{3.11}$$

$$\implies [a, b_i] \in \text{supp}((M_{13}, M_{23})|X = x, Y = y_1) \tag{3.12}$$

Denote $\text{supp}(M_{23}|M_{12} = m_{12}, Y = y_i)$ as B_i

$$\begin{aligned}
&Pr(M_{13} = a|X = x) = Pr(M_{13} = a|X = x, Y = y_1) \\
&\overset{(a)}{\geq} \sum_{i=1}^{|S_x|} \sum_{b \in B_i} Pr(M_{13} = a, M_{23} = b|X = x, Y = y_1) \\
&\overset{(b)}{=} \sum_{i=1}^{|S_x|} \sum_{b \in B_i} Pr(M_{13} = a, M_{23} = b|X = x, Y = y_i) \\
&= \sum_{i=1}^{|S_x|} \sum_{b \in B_i} \sum_{m \in \mathcal{M}_{12}} Pr(M_{12} = m)Pr(M_{13} = a|X = x, M_{12} = m)Pr(M_{23} = b|Y = y_i, M_{12} = m) \\
&\overset{(c)}{\geq} \sum_{i=1}^{|S_x|} \sum_{b \in B_i} Pr(M_{12} = m_{12})Pr(M_{13} = a|X = x, M_{12} = m_{12})Pr(M_{23} = b|Y = y_i, M_{12} = m_{12}) \\
&\overset{(d)}{=} |S_x|\, Pr(M_{12} = m_{12})Pr(M_{13} = a|X = x, M_{12} = m_{12})
\end{aligned}$$

where (a) is by marginalizing over all b in B_i, (b) is from (3.11), (c) is by assumption that $[a, b_i] \in \text{supp}((M_{13}, M_{23})|M_{12} = m_{12}, X = x, Y = y_i)$, and (d) is by marginalizing over all b in $B_i = \text{supp}(M_{23}|M_{12} = m_{12}, Y = y_i)$. Therefore we get the following, which we shall represent with a shorthand notation for the future

$$\begin{aligned}
Pr(M_{12} = m_{12}) &\leq \frac{Pr(M_{13} = a|X = x)}{|S_x|\, Pr(M_{13} = a|X = x, M_{12} = m_{12})} \\
p(m_{12}) &\leq \frac{p(a|x)}{|S_x|\, p(a|m_{12}, x)}
\end{aligned} \tag{3.13}$$

whenever $a \in \text{supp}(M_{13}|M_{12} = m_{12}, X = x)$. We proceed to bound the entropy as follows. Denote $\text{supp}(M_{13}|M_{12} = m, X = x)$ as $A(m, x)$ and $\text{supp}(M_{13}|X = x)$ as $A(x)$

$$
\begin{aligned}
H(M_{12}) &= \sum_{m \in \mathcal{M}_{12}} p(m) \log_2 \left(\frac{1}{p(m)} \right) \\
&= \sum_{m \in \mathcal{M}_{12}} \sum_{a \in A(m,x)} p(a|m,x) p(m) \log_2 \left(\frac{1}{p(m)} \right) \\
&\overset{(a)}{\geq} \sum_{m \in \mathcal{M}_{12}} \sum_{a \in A(m,x)} p(a|m,x) p(m) \log_2 \left(\frac{|S_x|\, p(a|m,x)}{p(a|x)} \right) \\
&= \sum_{a \in A(x)} \sum_{m \in \mathcal{M}_{12}} p(a|m,x) p(m) \log_2 \left(\frac{|S_x|}{p(a|x)} \right) - \sum_{m \in \mathcal{M}_{12}} \sum_{a \in A(m,x)} p(a|m,x) p(m) \log_2 \left(\frac{1}{p(a|m,x)} \right) \\
&= \log_2 |S_x| + \sum_{a \in A(x)} p(a|x) \log_2 \left(\frac{1}{p(a|x)} \right) - \sum_{m \in \mathcal{M}_{12}} p(m) H(M_{13}|M_{12} = m, X = x) \\
&= \log_2 |S_x| + H(M_{13}|X = x) - H(M_{13}|M_{12}, X = x) \\
&= I(M_{12}; M_{13}|X = x) + \log_2 |S_x|
\end{aligned}
$$

where (a) is from (3.13), and the rest follow from the usual properties of entropy, which completes the proof.

Notice that in the absence of private randomness, $I(M_{12}; M_{13}|X = x) = H(M_{13}|X = x)$ and thus we get the bound in (2.5). With private randomness, this can be a lower number. Since (3.9) holds for any choice of $x \in \mathcal{X}$, we can say

$$H(M_{12}) \geq \max_{x \in \mathcal{X}} I(M_{12}; M_{13}|X = x) + \log_2 |S_x| \tag{3.14}$$

For the AND function choosing $x = 0$ with $S_x = \{0, 1\}$, we get

$$H(M_{12}) \geq I(M_{12}; M_{13}|X = x) + 1 \tag{3.15}$$

We can evaluate the term in (3.9) for the protocol with private randomness, mentioned in Section 2.5. For the given protocol,

$$
\begin{aligned}
H(M_{13}|X = x) &= \log_2 6 && \text{for any } x \in \{0, 1, 2\} \\
H(M_{13}|M_{12}, X = x) &= \log_2 2 && \text{if } x = 2 \\
|S_x| &= 3 && \text{if } x = 2
\end{aligned}
$$

Using this, we get $H(M_{12}) \geq \log_2 9$. The protocol however uses $H(M_{12}) = \log_2 12$. We note that while the term in (3.9) is a correct bound for protocols with private randomness, it is not clear how to evaluate it without knowing the protocol a priori.

3.4 Relation with Distribution Design

Distribution design is a problem that has been studied in [6]. In this problem, the goal is to design a joint distribution on a set of random variables $X_1, \ldots, X_n$, such that a set of constraints are satisfied. It is of interest to study the minimum cardinality of the random variables required to satisfy these constraints. The constraints on the distribution may be of two types. Consider two sets of random variables of size d, $(X_{i_1}, \ldots, X_{i_d})$ and $(X_{i'_1}, \ldots, X_{i'_d})$. The first type of constraint is that the two sets of random variables are identically distributed, i.e. they have the same joint distribution (denoted by $\equiv$). The second type of constraint is that the two sets of random variables are disjoint, i.e. their supports are disjoint (denoted by $\|$).

The problem of distribution design was motivated from many cryptographic applications, including secret sharing, garbling schemes, and secure multiparty computation. Here, we show how this problem can be used to derive bounds for secure multiparty computation. In the three-party secure computation of AND, consider the random variables $(M^0_{13}, M^1_{13}, M^0_{23}, M^1_{23})$, where M^0_{13} is the message M_{13} sent by Alice when her input X is 0. Corresponding to the security conditions for AND (given in Section 1.3), we have the following constraints:

$$\begin{gathered}(M^0_{13}, M^0_{23}) \equiv (M^0_{13}, M^1_{23}) \equiv (M^1_{13}, M^0_{23}) \\ (M^1_{13}, M^1_{23}) \,\|\, (M^0_{13}, M^0_{23}) \qquad (M^1_{13}, M^1_{23}) \,\|\, (M^0_{13}, M^1_{23}) \qquad (M^1_{13}, M^1_{23}) \,\|\, (M^1_{13}, M^0_{23})\end{gathered} \tag{3.16}$$

We use the following lemma from [6]. In terms of notations, consider the given set of random variables indexed by $[n] = \{1, \ldots, n\}$. Share size denotes $\max_{i \in [n]} \lceil \log_2 |\mathrm{supp}(X_i)| \rceil$. A distribution design is *d-symmetric* if all permutations of $(X_{i_1}, \ldots, X_{i_d})$ are identically distributed, for all such sets of d random variables. A distribution design is *(d-1)-secret* if $(X_{i_1}, \ldots, X_{i_t}) \,\|\, (X_{i'_1}, \ldots, X_{i'_t})$ for any two such sets of $t \leq d-1$ random variables.

Lemma 3.4. *Let $A_0 \subseteq [n]$ be a set of size $1 < d < n$ and consider the constraints $\mathcal{R}_{A_0} = \{A \,\|\, A_0 \; : A \subseteq [n], |A| = d, A \neq A_0\} \cup \{A \equiv A' \; : \; A, A' \subseteq [n], |A| = |A'| = d, A, A' \neq A_0\}$. Then there exists a d-symmetric (d-1)-secret distribution design for $\mathcal{R}_{A_0}$ with share size at most $\min\{2d \cdot \log_2 n, n-1\}$.*

The proof of this lemma in [6] also outlines the construction of the distribution design. For our problem of three-party secure computation of AND, we have $n = 4$ and $d = 2$ with $A_0 = \{M_{13}^1, M_{23}^1\}$ according to (3.4), and this lemma guarantees the existence of a distribution design for this problem with $|\text{supp}(M_{13})|$ and $|\text{supp}(M_{23})|$ at most 2^3 (since $\text{supp}(M_{13}) = \text{supp}(M_{13}^0) = \text{supp}(M_{13}^1)$). For the multiparty setting, $n = 2k$ and $d = k$ and this lemma gives us that $|\text{supp}(M_i)|$ is at most 2^{2k-1} for $i = 1, \ldots, k$.

We find that this lemma for distribution design gives an upper bound of $2^{O(k)}$ for $|\text{supp}(M_i)|$ but for AND we have a protocol which achieves $O(k)$. Observe that in the case of three-party secure computation, the constraints used to evaluate Lemma 3.4 include extra constraints of the form $(M_{13}^0, M_{13}^1) \,\|\, (M_{13}^1, M_{23}^1)$, and $(M_{13}^0, M_{13}^1) \equiv (M_{13}^0, M_{23}^0)$. Both of these forms of constraints do not make sense in the context of our problem. Further, the proposed construction may not be optimal in terms of the share sizes.

Clearly, the problem of finding distributions of messages for secure multiparty computation is closely related to distribution design. In fact, secure multiparty computation of any binary output function can be posed as a distribution design problem. It is therefore of interest to study the problem of distribution design also, to give answers for both these problems. In the multiparty setting, let the input of each of the k parties be $X_i \in \mathcal{X}_i$, $i = 1, \ldots, k$ and the function $f : \mathcal{X}_1 \times \ldots \times \mathcal{X}_k \to \{0, 1\}$. Choose the random variables $\{M_i^x : x \in \mathcal{X}_i, i = 1, \ldots, n\}$. The constraints are of the form

$$\begin{aligned} (M_1^{x_1}, \ldots, M_k^{x_k}) &\equiv (M_1^{x_1'}, \ldots, M_k^{x_k'}) && \text{if } f(x_1, \ldots, x_k) = f(x_1', \ldots, x_k') \\ (M_1^{x_1}, \ldots, M_k^{x_k}) &\parallel (M_1^{x_1'}, \ldots, M_k^{x_k'}) && \text{if } f(x_1, \ldots, x_k) \neq f(x_1', \ldots, x_k') \end{aligned} \tag{3.17}$$

Chapter 4

Conclusion and Future Work

Secure multiparty computation is a problem with several relevant applications. However most of the currently known theoretical guarantees are not tight in general. This project studied a specific case of secure multiparty computation, for the function AND. In particular, this project showed that the randomness bounds derived for secure computation of AND, are tight under a larger class of protocols. Further, this project links the problem of secure multiparty computation of functions to another problem, that is distribution design.

While the bounds presented in this project are tight for the AND function, it is of interest to look at more generic bounds that can extend to other functions as well. In particular, the method used for this proof may be made more systematic and general. Further future work is in deriving information theoretic communication and randomness bounds. This involves finding how to evaluate the bound presented in (3.3) in terms of the inputs of the function, and extending the bounds in [4] to the multiparty setting.

Finally, it has been established that the secure multiparty computation problem can be formulated as a distribution design problem. The format in which this has been formulated does not explicitly use the common randomness variables. Thus the constraints in the distribution design problem may be modified to the context of this problem. We can also consider a variation of distribution design, that we can call support design. Here we are interested only in designing the support sets of the distribution, thus getting cardinality bounds on the random variables. As a support design problem, the constraints will be of the form as in (3.2)-(3.4).

Bibliography

[1] Uri Feige, Joe Killian, and Moni Naor. A minimal model for secure computation (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 554–563, New York, NY, USA, 1994. ACM. ISBN 0-89791-663-8. doi: 10.1145/195058.195408. URL http://doi.acm.org/10.1145/195058.195408.

[2] R Cramer, I.B. Damgård, and J.B. Nielsen. *Secure multiparty computation and secret sharing*. 01 2015. doi: 10.1017/CBO9781107337756.

[3] Rafi Heiman. Secure audio teleconferencing: A practical solution. In Rainer A. Rueppel, editor, *Advances in Cryptology — EUROCRYPT' 92*, pages 437–448, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. ISBN 978-3-540-47555-2.

[4] Deepesh Data, Vinod M. Prabhakaran, and Manoj M. Prabhakaran. Communication and randomness lower bounds for secure computation. *CoRR*, abs/1512.07735, 2015. URL http://arxiv.org/abs/1512.07735.

[5] Sundara Rajan S, S. Rajakrishnan, A. Thangaraj, and V. Prabhakaran. Lower bounds and optimal protocols for three-party secure computation. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1361–1365, July 2016. doi: 10.1109/ISIT.2016.7541521.

[6] Amos Beimel, Ariel Gabizon, Yuval Ishai, and Eyal Kushilevitz. Distribution design. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 81–92, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4057-1. doi: 10.1145/2840728.2840759. URL http://doi.acm.org/10.1145/2840728.2840759.