New Techniques for Analyzing Fully Secure Protocols: A Case Study of Solitary Output Secure Computation

Bar Alon* alonbar08@gmail.com
Benjamin Saldman† binyaminsaldman@gmail.com
Eran Omri† omrier@ariel.ac.il

March 19, 2025

Abstract

Solitary output secure computation allows a set of mutually distrustful parties to compute a function of their inputs such that only a designated party obtains the output. Such computations should satisfy various security properties such as correctness, privacy, independence of inputs, and even guaranteed output delivery. We are interested in full security, which captures all of these properties. Solitary output secure computation has been the study of many papers in recent years, as it captures many real-world scenarios.

A systematic study of fully secure solitary output computation was initiated by Halevi et al. [TCC 2019]. They showed several positive and negative results, however, they did not characterize what functions can be computed with full security. Alon et al. [EUROCRYPT 2024] considered the special, yet important case, of three parties with Boolean output, where the output-receiving party has no input. They completely characterized the set of such functionalities that can be computed with full security. Interestingly, they also showed a possible connection with the seemingly unrelated notion of *fairness*, where either all parties obtain the output or none of them do.

We continue this line of investigation and study the set of three-party solitary output Boolean functionalities where *all* parties hold private inputs. Our main contribution is defining and analyzing a family of "special-round" protocols, which generalizes the set of previously proposed protocols. Our techniques allow us to identify which special-round protocols securely compute a given functionality (if such exists). Interestingly, our analysis can also be applied in the *two-party* setting (where fairness is an issue). Thus, we believe that our techniques may prove useful in additional settings and deepen our understanding of the connections between the various settings.

*Department of Computer Science, Georgetown University.
†Department of Computer Science, Ariel University. Ariel Cyber Innovation Center (ACIC).

Contents

1 Introduction

Solitary output secure computation allows a set of mutually distrustful parties to compute a function of their inputs such that only a single party obtains the output. Such computation should preserve several security properties. These include correctness, privacy, independence of inputs, fairness, and even guaranteed output delivery. Full security captures all of these security properties.[1]

Solitary output secure computation appears in many real-life scenarios. For instance, consider a data analyst who wishes to perform statistics on users' data while maintaining the privacy of the data. Solitary output secure computation is further considered in many cryptographic settings, such as privacy-preserving federated learning [15, 13, 30], private simultaneous messages (PSM) protocols [22] and its robust version [11, 1], and in the setting of very large-scale computations for tech giants [6].

Solitary output secure computation also eliminates the need to achieve *fairness*, where, intuitively, it is required that either all parties obtain the output or none of them do. Indeed, fairness is not an issue in the case that only a single party obtains the output. This implies that the impossibility result of Cleve [17], which shows that achieving fairness is impossible in general without an honest majority, does not apply to the solitary output setting. This motivated Halevi et al. [27] to initiate a systematic study of solitary output secure computation. Although they showed several positive results, they also showed that there are solitary output functionalities that cannot be securely computed. However, their results do not characterize what functionalities can be securely computed. This motivated Alon et al. [7] to analyze the special case of three-party Boolean output functionalities, where the output-receiving party has no input. Interestingly, their results show that there might be a deep connection between fairness and solitary output secure computation. Indeed, both their techniques and results strongly resemble the techniques and results from the fairness literature [26, 8, 31, 9]. Thus, studying one setting could help better understand the other (seemingly unrelated) setting.

We continue this line of investigation, hoping to deepen our understanding of the connection between the two settings. Specifically, we ask the following question.

What solitary output functionalities can be computed with full security?
How is this related to fairness?

Assuming an honest majority, classical results show that any functionality can be computed with full security [24, 14, 34]. In the dishonest majority setting, Halevi et al. [27] showed there exists a solitary output functionality that cannot be securely computed, however, their results leave a gap between the positive and negative results. Alon et al. [7] considered the three-party setting, where the output-receiving party has no input. They provided a complete characterization of the set of such Boolean functionalities that can be securely computed. However, the case where the output-receiving party also has an input remained open.

1.1 Our Results

We consider the already challenging case of three parties with a Boolean output. Unlike in [7], we also let the output-receiving party hold an input. Our main contribution is an analysis of a natural generalization of the protocol of [7]. Interestingly, our analysis is also applicable to a natural

[1] Formally, security is defined using the real vs. ideal world paradigm, where security requires the real world to emulate an ideal world.

generalization of the two-party GHKL protocol [26, 9]. We believe our analysis can also be applied to other settings in addition to those specified. In addition to our positive result, we present a negative result, generalizing that of [7].

We next describe our results in more detail. Before doing so, following [4], we first introduce the *dealer model* for three-party solitary output computation,[2] which simplifies the proofs and descriptions of the protocols.

The dealer model. The dealer model is a middle ground between the real and ideal worlds. Here, similarly to the ideal world, the parties interact via a trusted dealer. However, the computation proceeds in rounds, and the (malicious) adversary can abort the computation. This model proves useful for simplifying the description of protocols and their proof of security. This is because the model allows us to abstract away the technicalities of implementing cryptographic primitives, which allows for protocols in this model to achieve information-theoretic security. Additionally, the model limits the adversary's capabilities, allowing it only to abort the protocol based on what it learned from the dealer (and change the inputs it sends to the dealer). This makes the security analysis of such protocols much simpler. Finally, we show that *any* protocol in the dealer model already satisfies several security properties (e.g., the output-receiving party cannot attack alone). This further simplifies the analysis. We refer the reader to Section 3.1 for more details.

We define a dealer model for the solitary output three-party setting and show how to compile a protocol from the dealer model into a real-world protocol and vice versa. Denote the parties by A, B, and C, and let C be the output-receiving party. Protocols in the dealer model are roughly defined as follows. First, the parties send their inputs to the trusted dealer. The dealer then computes *backup values* for each pair (A, C) and (B, C). This will later be used as the output of C in case party B or A, respectively, aborts. The dealer then proceeds to interact with the parties in a way that allows each pair to learn its backup values one after the other. The parties A and B can respond to the dealer with either $\mathsf{continue}$ or abort, indicating whether they can continue to the next round, or if the computation should halt and C should receive an output (which will be either the last backup value it received or a default value). We stress that an honest party always responds with $\mathsf{continue}$. We show that secure protocols in the dealer model can be compiled into a secure real-world protocol assuming oblivious transfer (OT). We further show that the converse also holds. That is, we show that secure protocols in the real world can be compiled into secure protocols in the dealer model.

Theorem 1.1 (Informal)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{W}$ be a solitary output three-party functionality. Then, assuming secure protocols for OT exist, f can be computed with full security in the real world if and only if it can be computed with full security in the dealer model.*

Having this result at hand, we can restrict the discussion to protocols in the dealer model. We formally provide the formal description of the model in Section 3 and prove the theorem.

Positive result for solitary output Boolean three-party functionalities. We identify a family of solitary output three-party functionalities that can be securely computed. Towards proving the result, we describe a general family of protocols, which follow the special-round paradigm [25, 26]. We then show that each such protocol is secure if a certain system of linear equations and

[2] The dealer model described in [4] is for the *friends and foes* setting, where the two dishonest parties are *not* colluding and one of them is semi-honest.

inequalities can be satisfied. Thus, we describe a collection of systems and show that if there exists a solvable system from our collection, then there exists a protocol from the family that securely computes a given functionality.

Theorem 1.2 (Informal, a sufficient condition)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a solitary output Boolean three-party functionality. Assume that secure protocols for OT exist. Then, there exists a family of systems of linear equations and inequalities, such that if any of them admit a solution, then f can be computed with full security.*

The formal description of the family of systems is given in Section 4. We note that our family of protocols generalizes the one constructed by [7] in a non-trivial way. We further show that a solution for the system of linear equations is necessary for such protocols (see Section 4.2 for more details).

We next show an example. Consider the following solitary output Boolean three-party functionality f described by the two matrices below.

$$\mathbf{M}_0 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{M}_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

We can think of the inputs of A and B as a subset of $\{1,2,3\}$ of size either 1 or 2, and of the input of C as either 0 or 1. Then, for $\mathcal{S}, \mathcal{T} \subseteq \{1,2,3\}$ such that $1 \leq |\mathcal{S}|, |\mathcal{T}| \leq 2$ it holds that $f(\mathcal{S}, \mathcal{T}, 0) = 1$ if $\mathcal{S} \cap \mathcal{T} = \emptyset$ and $f(\mathcal{S}, \mathcal{T}, 0) = 0$ if $\mathcal{S} \cap \mathcal{T} \neq \emptyset$, and $f(\mathcal{S}, \mathcal{T}, 1) = 1$ if $|\mathcal{S}| = |\mathcal{T}|$ and $f(\mathcal{S}, \mathcal{T}, 1) = 0$ if $|\mathcal{S}| \neq |\mathcal{T}|$.

We stress that the 2-ary functionality $f_0(x, y, \lambda) = f(x, y, 0)$ where λ is the empty string (i.e., where C's input is fixed to 0), was shown to be impossible to compute with full security [7]. However, we show that f can be computed with full security (see Example 4.7).

An analysis of the system of equations. Reducing the existence of a secure protocol to the existence of a solution to some system is unsatisfactory. Indeed, it is preferable to know when a secure protocol exists, rather than when a given protocol is secure. Therefore, we initiate an analysis of the family of systems that arise from Theorem 1.2. As this turns out to be very challenging, we consider only the system of linear equations (without the inequalities) and analyze when a solution for such a system exists. This, in turn, allows us to identify when one of the systems is solvable, thus giving a sufficient condition for the existence of a secure special-round protocol. Since our analysis is general enough, we believe that our techniques can be useful in analyzing other settings. To exemplify this, we show how to apply our analysis to the two-party setting for Boolean functionalities. Although a full characterization for this setting is already known [9], this shows that there might be a connection between the two settings, and perhaps other settings as well.[3]

To simplify the introduction, we describe our result for a special case of the system (the general case is given below in Section 1.2). Although it is a very limited result, it allows us to show the

[3] One may argue that the connection is only because we analyze the system that arises from the family of special-round protocols. However, the only fully secure protocols we are aware of are special-round protocols.

main techniques of our analysis. Fix a matrix $\mathbf{M}$ and a vector $\boldsymbol{\beta}$. Let $\mathbf{B}$ be the diagonal matrix whose entries are that of $\boldsymbol{\beta}$. Consider a system of the form

$$\mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}, \tag{1}$$

where $\mathbf{X}$ is the unknown of the system. Roughly speaking, $\mathbf{M}$ encodes a function, $\boldsymbol{\beta}$ encodes a protocol in the dealer model for computing the function (i.e., it encodes the distributions for the possible outputs of an honest C in case another party aborts in some round), and $\mathbf{X}$ encodes the strategy by which the simulators sample the inputs of the corrupted parties in the ideal world. In particular, Theorem 1.2 roughly states that a solution $\mathbf{X}$ implies the security of the protocol.[4] Thus, we ask the following question:

For what values of $\boldsymbol{\beta}$ there exists a solution $\mathbf{X}$ to System (1)?

To state our result, we first define an equivalence relation between two columns of $\mathbf{M}$. More concretely, we look at the *reduced row echelon form* of $\mathbf{M}$, and we are interested in when two columns have a non-zero entry at some (common) row in the reduced row echelon form of $\mathbf{M}$. Since such a relation is not transitive, we will consider its *transitive closure*.

Definition 1.3 (Informal, equivalence relation between columns)**.** *Given a matrix $\mathbf{M}$ over $\mathbb{R}$, we let $\mathbf{R}$ be its reduced row echelon form. For two columns y, y' in $\mathbf{R}$ we write $y \sim y'$ if there exists a row x in $\mathbf{R}$ such that $\mathbf{R}(x, y), \mathbf{R}(x, y') \neq 0$. Finally, we define the equivalence relation $\equiv$ to be the transitive closure of $\sim$. That is, $y \equiv y'$ if there exists a series of columns $y_1, \ldots, y_k$ such that*

$$y \sim y_1 \ldots y_k \sim y'.$$

Consider for example the following matrix $\mathbf{M}$

$$\mathbf{M} = \begin{pmatrix} 0 & 1/4 & 1/2 \\ 1/4 & 1/2 & 3/4 \\ 1/2 & 3/4 & 1 \end{pmatrix}.$$

The reduced row echelon form $\mathbf{R}$ of $\mathbf{M}$ is

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Observe that $y_1 \sim y_3$, and that $y_2 \sim y_3$. Then by transitivity, it also holds that $y_1 \equiv y_2$. Thus, all columns are equivalent.

We now state the result that characterizes when System (1) is solvable.

Theorem 1.4 (Informal, analysis of the system of equations)**.** *There exists a matrix $\mathbf{X}$ such that $\mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}$ if and only if for all columns y and y' in $\mathbf{M}$ such that $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$.*

For the example above, Theorem 1.4 asserts that a solution exists if and only if $\beta_1 = \beta_2 = \beta_3$. In Section 5 we consider the more general system that arises from the security requirements, and show when it admits a solution.

[4] Technically, Theorem 1.2 requires a solution for a more complicated system that also includes inequalities. We ignore this to simplify this introduction.

Application to fair two-party computation. We show how the above analysis can be applied to fairness in the two-party setting. In more detail, similarly to the solitary output setting, we generalize previous protocols [26, 9], which follow the "special-round" paradigm. We call these special-round protocols. We then analyze when a protocol from this family securely computes a given Boolean two-party functionality, by reducing the security properties to the existence of a solution to a system of linear equations (with no inequalities).

Interestingly, our result "almost" characterizes the set of special-round protocols that securely compute a given functionality. In the following, for a function f we let $\mathbf{M}_f$ denote its associated matrix.

Theorem 1.5 (Informal, almost characterization of special-round protocols in the two-party setting)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a Boolean two-party functionality with an associated matrix $\mathbf{M}_f$, and fix a vector $\boldsymbol{\beta}$. Then, if a secure protocol for oblivious transfer exists, and there exist a matrix $\mathbf{X}$ and a vector $\mathbf{p}$ over $\mathbb{R}$ such that*

$$\begin{cases} \mathbf{X} \cdot (\mathbf{M}_f || \mathbf{1}) = (\mathbf{M}_f || \mathbf{1}) \cdot \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \\ \mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T \\ \mathbf{p}^T \cdot \mathbf{1} = 1 \end{cases} .$$

Then there exists a special-round protocol that computes f with full security. Conversely, assume there exists a special round protocol that computes f with full security. Then there exist a matrix $\mathbf{X}$ and a vector $\mathbf{p}$ over $\mathbb{R}$ such that

$$\begin{cases} \mathbf{X} \cdot \mathbf{M}_f = \mathbf{M}_f \cdot \mathbf{B} \\ \mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T \\ \mathbf{p}^T \cdot \mathbf{1} = 1 \end{cases} .$$

We can now combine Theorem 1.5 and Theorem 1.4 to analyze when the matrix $\mathbf{X}$ exists for the above systems. Note that Theorem 1.5 "almost" characterizes the set of special-round protocols that securely compute a given Boolean two-party functionality. Indeed, the equations for $\mathbf{X}$ in first system can be written as $\mathbf{X} \cdot \mathbf{M}_f = \mathbf{M}_f \cdot \mathbf{B}$ and $\mathbf{X} \cdot \mathbf{1} = \mathbf{1}$. Thus, the only difference between the two systems is the latter condition for $\mathbf{X}$. We formally state and prove the theorem in Section 5.2.

Impossibility of computing strong semi-balanced functionalities. We generalize the impossibility result of Alon et al. [7] to include the case where the output-receiving party has an input. In more detail, [7] identified a class of solitary output Boolean functionalities that cannot be securely computed. They called this class *strong semi-balanced*. We first generalize the notion of strong semi-balanced functionalities to the case where the output-receiving party also holds an input. Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a Boolean solitary output Boolean three-party functionality. Assume for simplicity that f is deterministic. For $z \in \mathcal{Z}$ we let $\mathbf{M}_z$ be the matrix defined as $\mathbf{M}_z(x,y) = f(x,y,z)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We refer to the collection $\{\mathbf{M}_z\}_{z \in \mathcal{Z}}$ as the associated matrices of f. Strong semi-balanced functionalities are defined as follows.

Definition 1.6 (Informal, strong semi-balanced functionalities)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a solitary output Boolean three-party functionality, and let $\{\mathbf{M}_z\}_{z \in \mathcal{Z}}$ be its associated matrices. We*

call f strong semi-balanced if there exist two vectors $\mathbf{p}$ and $\mathbf{q}$ over $\mathbb{R}$, and there exists $z \in \mathcal{Z}$ such that for all x, y, and z' it holds that

$$\begin{cases} \mathbf{p}^T \cdot \mathbf{M}_z = \mathbf{1}^T, \\ \mathbf{1}^T \cdot \mathbf{p} < 1, \\ -1 + \mathbf{1}^T \cdot \mathbf{p} \leq \mathbf{p}^T \cdot \mathbf{M}_{z'}(\cdot, y) \leq 1 \end{cases} \quad \textit{and} \quad \begin{cases} \mathbf{M}_z \cdot \mathbf{q} = \mathbf{1}^T, \\ \mathbf{1}^T \cdot \mathbf{q} < 1, \\ -1 + \mathbf{1}^T \cdot \mathbf{q} \leq \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} \leq 1 \end{cases}.$$

Intuitively, the vectors $\mathbf{p}$ and $\mathbf{q}$ encode strategies for (A, C) and (B, C), respectively, for "locking" the output distribution of C when it has input z. That is, the vectors encode strategies for sampling inputs for A and B, and strategies for performing local operations on the output. If a pair plays with its strategy, then the output distribution of C is independent of the third party's input (i.e., it is locked).

The two inequalities limit the amount of information an ideal-world simulator (corrupting C and an additional party) can obtain from the trusted party on the input of the honest party (assuming it sampled its input according to its strategy). We stress that the vectors $\mathbf{p}$ and $\mathbf{q}$ might contain non-negative entries.

Theorem 1.7 (Informal, impossibility for strong semi-balanced functionalities)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a strong semi-balanced solitary output three-party functionality. Then, f cannot be computed with full security.*

The formal proof appears in Section 6. We next give an example. Consider the following solitary output Boolean three-party functionality f whose associated matrices are

$$\mathbf{M}_0 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Similarly to the example of the possibility result, we can think of the inputs of A and B as a subset of $\{1,2,3\}$ of size between 1 to 2, and of the input of C as 0 or 1. Then, for $\mathcal{S}, \mathcal{T} \subseteq \{1,2,3\}$ such that $1 \leq |\mathcal{S}|, |\mathcal{T}| \leq 2$ it holds that $f(\mathcal{S}, \mathcal{T}, 0) = 1$ if $\mathcal{S} \cap \mathcal{T} = \emptyset$ and $f(\mathcal{S}, \mathcal{T}, 0) = 0$ if $\mathcal{S} \cap \mathcal{T} \neq \emptyset$, and it also holds that $f(\mathcal{S}, \mathcal{T}, 1) = 1$ if $\mathcal{S} = \mathcal{T}$ and $f(\mathcal{S}, \mathcal{T}, 1) = 0$ if $\mathcal{S} \neq \mathcal{T}$.

Note that the functionality $f_0(x, y) = f(x, y, 0)$ (i.e., where C's input is fixed to 0) was shown to be impossible to compute with full security [7]. However, their result is not applicable in our setting. Intuitively, this is because the ideal world simulator that controls C can change its input, which might help him to simulate. Our result essentially shows that this does not help the simulator.

We next show that f is a strong semi-balanced functionality, thus by Theorem 1.7, it cannot be computed securely. Let

$$\mathbf{p} = \mathbf{q} = (1, 1, 1, -1, -1, -1)^T.$$

Then, it holds that $\mathbf{p}^T \cdot \mathbf{M}_0 = \mathbf{1}^T$, and that $\mathbf{M}_0 \cdot \mathbf{q} = \mathbf{1}^T$. Additionally, note that the sum of the entries in $\mathbf{p}$ and $\mathbf{q}$ is strictly less than 1. Finally, it holds that

$$-1 \leq \mathbf{p}^T \cdot \mathbf{M}_1(\cdot, \mathcal{T}) \leq 1 \quad \text{and that} \quad -1 \leq \mathbf{M}_1(\mathcal{S}, \cdot) \cdot \mathbf{q} \leq 1$$

for all $\mathcal{T}, \mathcal{S} \subseteq \{1,2,3\}$ such that $1 \leq |\mathcal{T}|, |\mathcal{S}| \leq 2$. Therefore, f is strong semi-balanced, hence by Theorem 1.7 it cannot be computed with full security.

1.2 Our Techniques

We now describe our techniques. We begin by describing the construction of the dealer model. Throughout the rest of this section, we denote the parties by A, B, and C, their inputs by x, y, and z, respectively, and we let C be the output-receiving party.

The dealer model. We show that a solitary output Boolean three-party functionality f can be computed with full security in the real world (against malicious adversaries) if and only if it can be computed with full security in the dealer model (against fail-stop adversaries). We start by describing the interaction in the dealer model. An r-round protocol in the dealer model is defined as follows. First, all parties send their input to the dealer. Then, the dealer computes the *backup values* $a_0, \ldots, a_r$ and $b_0, \ldots, b_r$ (recall that the i^{th} backup value determines the output of the honest party in case a party aborts at round i) such that a_0 and b_0 do not depend on the inputs of A and B respectively. The dealer then needs to send these values to the parties. It does so by sharing the backup values it computed in a 2-out-of-2 secret-sharing scheme, and sends C it shares of $a_0, \ldots, a_r$ and $b_0, \ldots, b_r$. Then, for $i = 1$ to r the dealer does the following.

1. Approach party A and send its share $a_i[\mathsf{A}]$. Party A then responds with either continue or abort.
2. If A responds with abort, then approach party B that also responds with either continue or abort.
   - If B responds with continue, then send b_{i-1} to C and halt. Party C then outputs b_{i-1}.
   - If B responds with abort, then send a default value to C and halt. Party C then outputs the default value.
3. If A responds with continue, then approach party B and send its share $b_i[\mathsf{B}]$. Party B then responds with either continue or abort.
4. If B responds with abort, then approach party A that also responds with either continue or abort.
   - If A responds with continue, then send a_i to C and halt. Party C then outputs a_i.
   - If A responds with abort, then send a default value to C and halt. Party C then outputs the default value.

If no party sent abort, then the dealer sends $a_r = f(x, y, z)$ to C and halts. Party C then outputs $f(x, y, z)$.

We next show how to eliminate the dealer from the computation. More concretely, the parties share their backup values using a secret sharing scheme, and sign the shares using a signature scheme. This procedure is done using a secure-with-identifiable-abort protocol (i.e., the adversary can abort the execution and prevent the honest part from learning the output at the cost of revealing its identity). If A or B abort at the secure-with-identifiable-abort phase, then the remaining honest party computes a function of its and C's input, and C outputs the result. If no abort occurred, then in every round $i = 1$ to r each party A and B broadcast their share $b_i[\mathsf{A}]$ and $a_i[\mathsf{B}]$ and the corresponding their signatures respectively. We stress that this process is done one after the other. If the wrong share was sent (i.e., the verification failed) or a party aborted, then the remaining

party helps C to reconstruct the last backup value they can reconstruct by sending to C its share of the last backup value it knows. We refer the reader to Section 3 for more details.

It remains to describe how we compile a real-world protocol into a dealer model protocol. For that, the dealer samples randomness for each party and runs the real-world protocol in its head. For every round $i \in \{0, \ldots, r\}$ it computes the output of C in case party A or B aborted the computation at round i. It then sets the backup values for every round i as the values it computed in its head. See Section 3 for more details.

Overview of the positive result. For the positive result, for a given solitary output Boolean three-party functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$, we construct a family of systems of linear equalities and inequalities and show that if at least one system of the family can be satisfied, then f can be computed with full security. We derive the systems by presenting a family of protocols for computing such functionalities with full security and reducing the security properties into a system of linear equalities and inequalities.

We next describe the family of protocols. Inspired by previous works [26, 9, 7], the protocols follow the "special-round" paradigm. Loosely speaking, the parties sample together a special random round i^* that is unknown to any strict subset of them. Before round i^* is reached, the backup values of (A, C) and (B, C) are sampled at random and independently. After round i^* is reached these backup values are equal to the output (i.e., $f(x, y, z)$). Asharov et al. [9] were the first to modify this approach in the two-party setting. In their protocol, if A aborts in round i^* then B outputs a *constant* that is either 0 or 1. Crucially, this constant depends only on the functionality and not the inputs. This approach helped them to characterize the set of Boolean two-party functionalities that can be securely computed. Alon et al. [7] used a more general idea in the solitary output three-party setting. Specifically, if A aborts in round i^* then the output of C will be from a constant distribution (rather than a constant value) that depends only on the functionality. This allowed them to characterize the set of solitary output Boolean three-party functionalities where the output-receiving party does not hold an input.

As we show below, in our setting where C holds an input, the protocol of Alon et al. [7] will not be secure for some functionalities. That is, there exists a functionality that can be computed with full security, but only if the backup value b_{i^*-1} depends on the input of B. Thus, we introduce a family of protocols, where each is parametrized by a vector $\boldsymbol{\beta} = (\beta_{y,z})_{y \in \mathcal{Y}, z \in \mathcal{Z}} \in [0,1]^{|\mathcal{Y}| \cdot |\mathcal{Z}|}$.[5] Each $\beta_{y,z}$ is the probability that C outputs 1 if A aborts in round i^*, on inputs y and z for B and C, respectively.

We now describe a special-round protocol in more detail. Following Theorem 1.1, we may do so in the dealer model. Recall that a protocol in the dealer model is defined by the backup values. Given a round $i < i^* - 1$, we let $a_i = f(x, \widetilde{y}_i, z)$, where $\widetilde{y}_i \leftarrow \mathcal{Y}$, and we let $b_i = f(\widetilde{x}_i, y, z)$, where $\widetilde{x}_i \leftarrow \mathcal{X}$. If $i = i^* - 1$, we let $a_i = f(x, \widetilde{y}_i, z)$, where $\widetilde{y}_i \leftarrow \mathcal{Y}$, and we set $b_i = 1$ with probability $\beta_{y,z}$ and $b_i = 0$ with probability $1 - \beta_{y,z}$. Finally, if $i \geq i^*$, then we let $a_i = b_i = f(x, y, z)$. Reinterpreting the protocols of previous works, in the protocol of [7] $\beta_{y,z}$ is set to be independent of y, in the two-party protocol of [9] $\beta_{y,z}$ is set to be in $\{0,1\}$, and in the two-party protocol of [26, 8] $\beta_{y,z}$ is defined as $\beta_{y,z} = \Pr[f(\widetilde{x}, y) = 1]$ where $\widetilde{x} \leftarrow \mathcal{X}$.

To demonstrate the need for $\boldsymbol{\beta}$ to depend on both y and z, consider the equality functionality

[5] Formally, the protocol is also parametrized by another real number $\alpha \in (0,1]$ that corresponds to the distribution of the special round i^*. See Section 4 for more details.

$\mathsf{eq} : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ defined as

$$\mathsf{eq}(x,y,z) = \begin{cases} 1 & \text{if } x = y = z \\ 0 & \text{otherwise} \end{cases}.$$

We argue that $\boldsymbol{\beta}$ that is independent of y results in an insecure protocol. Specifically, a direct calculation shows that an adversary that corrupts *only* A and always aborts in the first round (without learning a_1) cannot be simulated. We formalize the argument in Proposition 4.5. We further observe that if we set $\beta_{y,z} = \Pr[f(\widetilde{x}, y, z) = 1]$ for $x \leftarrow \mathcal{X}$, then the protocol is secure. We refer to Section 4 for more detail.

It is left to analyze the security of the protocol. Similarly to previous works [26, 8, 31], we show that the protocol is secure if a certain system of linear equalities and inequalities has a solution.[6] Specifically, we obtain the following system. Let $\mathbf{M}_{\mathsf{r}}$ be the concatenation of all the associated matrices $\mathbf{M}_z$ by rows, and for every $z \in \mathcal{Z}$ let $\mathbf{B}_z$ be the diagonal matrix whose diagonal consists of $(\beta_{y,z})_{y\in\mathcal{Y}}^T$. We show that the protocol is secure if for all $z \in \mathcal{Z}$, there exists matrices $\mathbf{X}_z^0$ and $\mathbf{X}_z^1$ and there exists a vector $\mathbf{p}$ over $\mathbb{R}$ such that

$$\begin{cases} \mathbf{p} \cdot \mathbf{M}_z = (\beta_{y,z})_{y\in\mathcal{Y}}^T \\ \mathbf{p}^T \cdot \mathbf{1} = 1 \\ \mathbf{X}_z^1 \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{M}_z \cdot \mathbf{B}_z \\ \mathbf{X}_z^0 \cdot (\mathbf{J} - \mathbf{M}_{\mathsf{r}}) = (\mathbf{J} - \mathbf{M}_z) \cdot (\mathbf{I} - \mathbf{B}_z) \\ \mathbf{0} \le \mathbf{X}_z^w \cdot \mathbf{1} \le \mathbf{1}, \text{ for all } w \in \{0,1\} \\ \mathbf{X}_z^w(x,(x',z')) \ge 0 \text{ for all } x, x' \in \mathcal{X},\ z' \ne z, \text{ and } w \in \{0,1\} \end{cases},$$

where $\mathbf{J}$ is the all-one matrix (of appropriate dimensions) and $\mathbf{I}$ is the identity matrix. These constraints are obtained by defining a simulator for any adversary in the dealer model and comparing them to ensure security. Since the simulators can only change their inputs, they can be completely described using probability vectors (i.e., vectors whose entries are non-negative and whose entries sum to 1). These are encoded in the above system using the matrices $\{\mathbf{X}_z^w\}_{z\in\mathcal{Z},w\in\{0,1\}}$ and the vector $\mathbf{p}$.[7] Since the matrices $\{\mathbf{B}_z\}_{z\in\mathcal{Z}}$ encode a protocol in the dealer model, identifying for what collection $\{\mathbf{B}_z\}_{z\in\mathcal{Z}}$ a solution exists would imply which special-round protocols are secure. The formal proof can be found in Section 4.1.

Overview of the analysis of the system of equations. We next demonstrate our techniques for analyzing the system of linear equations. To simplify the introduction, we consider the following special case. Let $\mathbf{M} \in \mathbb{R}^{m\times\ell}$ and $\boldsymbol{\beta} \in \mathbb{R}^{\ell}$. Define $\mathbf{B} \in \mathbb{R}^{\ell\times\ell}$ to be the diagonal matrix whose entries on the main diagonal are the values in $\boldsymbol{\beta}$. Following the security requirements of our protocols, we consider the system

$$\mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}, \tag{2}$$

[6] We stress that in the two-party setting, there are only equations. See Section 5.2.

[7] Note that the columns in $\mathbf{X}_z^0$ and $\mathbf{X}_z^1$, and the vector $\mathbf{p}$ are *not* probability vectors. The choice of encoding the simulator in this way is because encoding directly using probability vectors results in a more complicated system with additional constraints.

where the matrix $\mathbf{X} \in \mathbb{R}^{m \times \ell}$ consists of the unknowns. Recall that our goal is to characterize the set of all $\boldsymbol{\beta}$'s for which a solution to Equation (2) exists. We prove that $\mathbf{X}$ exists if and only if for all $y \equiv y'$ (as in Definition 1.3) it holds that $\beta_y = \beta_{y'}$.

We first show how to simplify the system. Let $\mathbf{R}$ be the reduced row echelon form of $\mathbf{M}$. We show that there exists a solution $\mathbf{X}$ to Equation (2) if and only if there exists $\tilde{\mathbf{X}}$ such that

$$\tilde{\mathbf{X}} \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{B}.$$

To see this, first observe that since $\mathbf{R}$ resulted by performing elementary row operations on $\mathbf{M}$, there exists an invertible matrix $\mathbf{E}$ such that $\mathbf{E} \cdot \mathbf{M} = \mathbf{R}$. Multiplying both sides of Equation (2) by $\mathbf{E}$ from the left side results in

$$\mathbf{E} \cdot \mathbf{X} \cdot \mathbf{E}^{-1} \cdot \mathbf{E} \cdot \mathbf{M} = \mathbf{E} \cdot \mathbf{M} \cdot \mathbf{B}.$$

Let $\tilde{\mathbf{X}} = \mathbf{E} \cdot \mathbf{X} \cdot \mathbf{E}^{-1}$. Then a solution to Equation (2) exists if and only if there exists $\tilde{\mathbf{X}}$ such that

$$\tilde{\mathbf{X}} \cdot \mathbf{E} \cdot \mathbf{M} = \tilde{\mathbf{X}} \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{B}.$$

We argue that the system can be further simplified by removing rows x such that $\mathbf{R}(x, \cdot)$ is the all-zero vector. Indeed, for such a row the right-hand side is 0, hence we can always set $\tilde{\mathbf{X}}(x, \cdot)$ to be the all-zero vector. We conclude that there exists a solution to Equation (2) if and only if there exists $\hat{\mathbf{X}}$ such that

$$\hat{\mathbf{X}} \cdot \hat{\mathbf{R}} = \hat{\mathbf{R}} \cdot \mathbf{B}, \tag{3}$$

where $\hat{\mathbf{R}}$ is the matrix $\mathbf{R}$ with the rows of zeros removed. Note that since $\hat{\mathbf{R}}$ is the matrix $\mathbf{R}$ with the zero-rows removed, it holds that if a solution $\hat{\mathbf{X}}$ for Equation (3) exists, then it is unique. In what follows, we let m' denote the number of rows in $\hat{\mathbf{R}}$.

We now show that there exists a solution to Equation (3) if and only if for all $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$. For the first direction, assume that for all $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$. A column $y \in [\ell]$ is called a *pivot* in $\hat{\mathbf{R}}$ if there exists a row $x \in [m']$ such that $\hat{\mathbf{R}}(\cdot, y) = \mathbf{e}_x$ and for all $y' < y$ it holds that $\hat{\mathbf{R}}(\cdot, y') \neq \mathbf{e}_x$, where $\mathbf{e}_x$ is the x^{th} standard basis vector. Let $\mathsf{ptr}(\cdot)$ denote the bijection that for a pivot column y returns the unique row x such that $\hat{\mathbf{R}}(\cdot, y) = \mathbf{e}_x$. We show that the unique solution $\hat{\mathbf{X}}$ is given by

$$\hat{\mathbf{X}}(x, x') = \beta_{\mathsf{ptr}^{-1}(x')} \cdot \hat{\mathbf{R}}(x, \mathsf{ptr}^{-1}(x')),$$

for all $x, x' \in [m']$.

Fix $x \in [m']$ and consider the x^{th} row of the system. Then, for every column $y \in [\ell]$ it holds that

$$\hat{\mathbf{X}}(x, \cdot) \cdot \hat{\mathbf{R}}(\cdot, y) = \sum_{x' \in [m']} \hat{\mathbf{X}}(x, x') \cdot \hat{\mathbf{R}}(x', y) = \sum_{x' \in [m']} \beta_{\mathsf{ptr}^{-1}(x')} \cdot \hat{\mathbf{R}}(x, \mathsf{ptr}^{-1}(x')) \cdot \hat{\mathbf{R}}(x', y).$$

Now, if y is a pivot column then there exists exactly one row x' for which it holds that $\hat{\mathbf{R}}(x', y) = 1$ and for all other $x'' \neq x'$ it holds that $\hat{\mathbf{R}}(x'', y) = 0$. Thus, for such y the right-hand side equals $\beta_y \cdot \hat{\mathbf{R}}(x, y) = \hat{\mathbf{R}}(x, \cdot) \cdot \mathbf{B}(\cdot, y)$, as required.

Assume now that y is a free column in $\hat{\mathbf{R}}$. Let $\mathcal{P} \subseteq [\ell]$ denote the set of pivot columns in $\hat{\mathbf{R}}$. Since ptr is a bijection it follows that

$$\begin{aligned}\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) &= \sum_{x'\in[m']} \beta_{\mathsf{ptr}^{-1}(x')}\cdot\hat{\mathbf{R}}(x,\mathsf{ptr}^{-1}(x'))\cdot\hat{\mathbf{R}}(x',y)\\ &= \sum_{y_{\mathsf{piv}}\in\mathcal{P}} \beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y_{\mathsf{piv}})\cdot\hat{\mathbf{R}}(\mathsf{ptr}(y_{\mathsf{piv}}),y).\end{aligned}$$

Now, observe that there exists exactly one pivot column y_{piv} for which it holds that $\hat{\mathbf{R}}(x,y_{\mathsf{piv}}) \neq 0$. This is because ptr is a bijection. Hence, there exists at most one column y_{piv} for which it holds that $\hat{\mathbf{R}}(x,y_{\mathsf{piv}}) \neq 0$ and $\hat{\mathbf{R}}(x,y) \neq 0$. Therefore

$$\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = \beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y_{\mathsf{piv}})\cdot\hat{\mathbf{R}}(\mathsf{ptr}(y_{\mathsf{piv}}),y).$$

Furthermore, it holds that $x = \mathsf{ptr}(y_{\mathsf{piv}})$, hence

$$\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = \beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y_{\mathsf{piv}})\cdot\hat{\mathbf{R}}(x,y).$$

Now, if $\hat{\mathbf{R}}(x,y) = 0$ then $\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = 0$ and $\hat{\mathbf{R}}(x,y)\cdot\mathbf{B}(x,y) = 0$, hence they are equal. Otherwise, it follows that $y_{\mathsf{piv}} \sim y$. Since $\hat{\mathbf{R}}(x,y_{\mathsf{piv}}) = 1$, it follows that $\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = \beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y)$. As we assume that for all $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$, it also holds that $\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = \beta_y\cdot\hat{\mathbf{R}}(x,y) = \hat{\mathbf{R}}(x,\cdot)\cdot\mathbf{B}(\cdot,y)$.

The second direction is implied by the first one. First, note that for any y for which $\hat{\mathbf{R}}(\cdot,y) = \mathbf{0}$, it holds that y is not equivalent to any other y'. Thus, the constraints on β_y hold vacuously. For any other y, either it is a pivot column, or $y \sim y_{\mathsf{piv}}$ for some pivot column y_{piv}. We next show that for all free columns y, it holds that $\beta_y = \beta_{y_{\mathsf{piv}}}$. By an inductive argument, this implies the result.

Fix a free column y and a pivot column y_{piv} such that $y \sim y_{\mathsf{piv}}$. Since the solution $\hat{\mathbf{X}}$ is unique, as we showed in the first direction of the proof, it must be of the form

$$\hat{\mathbf{X}}(x,x') = \beta_{\mathsf{ptr}^{-1}(x')}\cdot\hat{\mathbf{R}}(x,\mathsf{ptr}^{-1}(x')), \tag{4}$$

for all $x, x' \in [m']$. Now, since we assume that $\hat{\mathbf{X}}$ solves the system, it must hold that $\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = \beta_y\cdot\hat{\mathbf{R}}(x,y)$ for all $x \in [m]$ and $y \in [\ell]$. By Equation (4),

$$\hat{\mathbf{X}}(x,\cdot)\cdot\hat{\mathbf{R}}(\cdot,y) = \sum_{y_{\mathsf{piv}}\in\mathcal{P}} \beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y_{\mathsf{piv}})\cdot\hat{\mathbf{R}}(\mathsf{ptr}(y_{\mathsf{piv}}),y).$$

Now, as discussed in the previous direction of the proof, the sum on the right-hand side equals $\beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y_{\mathsf{piv}})\cdot\hat{\mathbf{R}}(x,y)$. Since $\hat{\mathbf{R}}(x,y_{\mathsf{piv}}) = 1$ and $\hat{\mathbf{R}}(x,y) \neq 0$, it follows that

$$\beta_y\cdot\hat{\mathbf{R}}(x,y) = \beta_{y_{\mathsf{piv}}}\cdot\hat{\mathbf{R}}(x,y_{\mathsf{piv}})\cdot\hat{\mathbf{R}}(x,y)$$

implying that $\beta_y = \beta_{y_{\mathsf{piv}}}$.

We formally prove both directions of the theorem in Section 5.2.

Overview of the impossibility result. For the impossibility result, fix a strong-semi balanced solitary output Boolean three-party functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$. Then, by assumption there exist vectors $\mathbf{p}$ and $\mathbf{q}$ and there exists $z \in \mathcal{Z}$, such that for every $z' \in \mathcal{Z}$, for every $x \in \mathcal{X}$, and for every $y \in \mathcal{Y}$, it holds that

$$\begin{cases} \mathbf{p}^T \cdot \mathbf{M}_z = \delta_1 \cdot \mathbf{1}^T, \text{ where } \delta_1 > 0 \\ \mathbf{1}^T \cdot \mathbf{p} < \delta_1, \\ \sum_{x \in \mathcal{X}} |p_x| = 1, \\ -\delta_1 + \mathbf{1}^T \cdot \mathbf{p} \leq \mathbf{p}^T \cdot \mathbf{M}_{z'}(\cdot, y) \leq \delta_1 \end{cases} \text{ and } \begin{cases} \mathbf{M}_z \cdot \mathbf{q} = \delta_2 \cdot \mathbf{1}, \text{ where } \delta_2 > 0 \\ \mathbf{1}^T \cdot \mathbf{q} < \delta_2, \\ \sum_{y \in \mathcal{Y}} |q_y| = 1, \\ -\delta_2 + \mathbf{1}^T \cdot \mathbf{q} \leq \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} \leq \delta_2 \end{cases}.$$

Alon et al. [7] showed that if C doesn't hold an input then for every real-world protocol there exists an adversary that can attack the protocol and guess a certain property associated with the input of the honest party. We use the same technique of Alon et al. [7] by constructing an adversary that fixes the input of C to be z and works as the adversary defined by [7]. We stress that fixing the input of C is not enough for the attack to succeed as in the ideal world the simulator controlling C can change the input it sends to the trusted party. For that, we require the third constraint of f to hold for every $z' \in \mathcal{Z}$. We refer the reader to Section 6 for more details and formal arguments.

1.3 Related Works

The celebrated work of Cleve [17] showed that fair two-party coin-tossing is impossible to achieve. This implies the impossibility of computing fairly functions that imply coin-tossing such as XOR. Surprisingly, Gordon et al. [26] showed there is a non-trivial two-party functionality (i.e., it contains an embedded XOR) that can be securely computed. Their protocol follows the "special-round" paradigm, which we analyze in our paper. The work of Gordon et al. [26] was generalized [8, 31, 9]. In particular, Asharov et al. [9] characterized the set of symmetric two-party Boolean functionalities (where both parties output the same bit) that can be computed fairly. Finally, Daza and Makriyannis [21], Makriyannis [32] further generalize the results to include non-Boolean and asymmetric functionalities (i.e., the parties might obtain different functions applied to their inputs).

In the multiparty setting, Rabin and Ben-Or [34] showed that if there is an honest majority, and the parties are connected with secure point-to-point channels and are given access to a broadcast channel, then full security can be obtained without any cryptographic assumptions. Cohen and Lindell [18] studied the relation between fairness and guaranteed output delivery in the multiparty setting. They showed that any functionality that can be computed with guaranteed output delivery using a broadcast channel can be computed with fairness over point-to-point channels. Secure multiparty computation without broadcast was studied by [19, 5, 3]. In particular, [3] considered the solitary output three-party setting without broadcast, and completely characterized the case where the output-receiving party has no input, and the case where the output is one of three values (and the output-receiving party might have an input).

Gordon and Katz [25] were the first to consider the possibility of obtaining full security in the multiparty setting without an honest majority. They showed that the three-party majority functionality and n-party OR can be computed with full security. The case where exactly half of the parties are corrupted was considered by [9]. Finally, Dachman-Soled [20] considered the setting of a non-constant number of parties.

Relevant to our positive results, Lindell and Rabin [29] showed that protocols that admit a *committal round* (i.e., a fixed round by which the parties are committed to their inputs) could

not be fully secure without an honest majority. Note that special round protocols do not have a committal round.

1.4 Organization

In Section 2, we provide the notations and model of computation. In Section 3, we describe the dealer model for solitary output three-party functionalities. We also prove it is equivalent to the real model, and we prove some simple security properties. In Section 4, we introduce the family of special-round protocols, and we reduce their security to the solvability of a system of linear constraints. In Section 4.2, we show a necessary condition for the security of the protocols. In Section 5, we provide the full analysis of the system of linear equations and show an application to the two-party setting. Finally, in Section 6 we state and prove our impossibility result for securely computing strong semi-balanced functionalities.

2 Preliminaries

2.1 Notations

We use calligraphic letters to denote sets, uppercase for random variables and distributions, lowercase for values, and we use bold characters to denote vectors. For $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$. For a set $\mathcal{S}$ we write $s \leftarrow \mathcal{S}$ to indicate that s is selected uniformly at random from $\mathcal{S}$. PPT is short for probabilistic polynomial time. We let λ denote the empty string.

A function $\mu(\cdot)$ is negligible if for every positive polynomial $p(\cdot)$ there exists an N such that for every $n > N$ it holds that $\mu(n) < 1/p(n)$. We will write neg for an unspecified negligible function. For a randomized function (or an algorithm) f we write $f(x)$ to denote the random variable induced by the function on input x, and write $f(x; r)$ to denote the value when the randomness of f is fixed to r.

In this paper, all vectors are column vectors over $\mathbb{R}$. We denote by $\mathbf{1}_n$ and $\mathbf{0}_n$ the all-one and all-zero vectors of dimension n, respectively. We will remove n when its value is clear from the context. For $i \in [n]$ and vector $\mathbf{v} \in \mathbb{R}^n$ we denote the i^{th} entry in $\mathbf{v}$ by $v(i)$ or v_i. Let $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ be two vectors. We say that $\mathbf{v} \leq \mathbf{w}$ if for every $i \in [n]$ it holds that $v(i) \leq w(i)$. A vector $\mathbf{p} \in \mathbb{R}^n$ is called a *probability vector* if for every $i \in [n]$ it holds that $0 \leq p(i) \leq 1$, and $\sum_{i=1}^{n} p(i) = 1$. For such vectors, we write $v \leftarrow \mathbf{p}$ to indicate that v is sampled according to the distribution that corresponds to $\mathbf{p}$, namely, $v = i$ with probability $p(i)$. We will sometimes denote coordinates for a vector $\mathbf{v}$ with a pair of integers (i, j). We will write $v(i, j)$ instead of $v((i, j))$. An *affine combination* is a linear combination where the sum of the coefficients is 1. Finally, we let $|\mathbf{v}|$ denote the vector that is created by taking the absolute value in every entry of $\mathbf{v}$, i.e., its i^{th} position is $|v_i|$.

For a matrix $\mathbf{M} \in \mathbb{R}^{n \times m}$, we let $\mathbf{M}(\cdot, i)$ denote the i^{th} column of $\mathbf{M}$, and let $\mathbf{M}(i, \cdot)$ denote the i^{th} row. We denote its transpose by $\mathbf{M}^T$, its image by $\mathrm{Im}(\mathbf{M})$, and its kernel by $\ker(\mathbf{M})$. For two matrices $\mathbf{M}$ and $\mathbf{N}$ we let $(\mathbf{M}||\mathbf{N})$ be the matrix obtained by concatenating them by columns. We further denote by $(\mathbf{M}||_{\mathsf{r}}\mathbf{N})$ the concatenation of the matrices by rows. That is, $(\mathbf{M}||_{\mathsf{r}}\mathbf{N}) = (\mathbf{M}^T||\mathbf{N}^T)^T$. Finally, we let $\mathbf{J}_{n,m}$ be the all-ones matrix of size $n \times m$. We remove n and m when their value is clear from the context.

A *distribution ensemble* $X = \{X(a, n)\}_{a \in \mathcal{D}_n, n \in \mathbb{N}}$ is an infinite sequence of random variables indexed by $a \in \mathcal{D}_n$ and $n \in \mathbb{N}$, where $\mathcal{D}_n$ is a domain that might depend on n. We define computational indistinguishability and statistical distance as follows.

Definition 2.1 (Computational indistinguishability)**.** *Let* $X = \{X(a,n)\}_{a\in\mathcal{D}_n,n\in\mathbb{N}}$ *and* $Y = \{Y(a,n)\}_{a\in\mathcal{D}_n,n\in\mathbb{N}}$ *be two distribution ensembles. We say that* X *and* Y *are computationally indistinguishable, denoted* $X \overset{\text{C}}{\equiv} Y$*, if for every non-uniform polynomial-time algorithm* D*, there exists a negligible function* $\mu(\cdot)$ *such that for every* $a \in \mathcal{D}_n$ *and* $n \in \mathbb{N}$ *it holds that*

$$|\Pr[\mathsf{D}(X(a,n)) = 1] - \Pr[\mathsf{D}(Y(a,n)) = 1]| \leq \mu(n).$$

Definition 2.2 (Statistical distance)**.** *The statistical distance between two random variables* X *and* Y *is defined as*

$$\mathrm{SD}(X(a,n), Y(a,n)) = \frac{1}{2} \cdot \sum_s |\Pr[X(a,n) = s] - \Pr[Y(a,n) = s]|.$$

If $X = \{X(a,n)\}_{a\in\mathcal{D}_n,n\in\mathbb{N}}$ *and* $Y = \{Y(a,n)\}_{a\in\mathcal{D}_n,n\in\mathbb{N}}$ *are two distribution ensembles, we say that* X *and* Y *are statistically close, denoted* $X \overset{\text{S}}{\equiv} Y$*, if for all* $a \in \mathcal{D}_n$ *and* $n \in \mathbb{N}$ *it holds that*

$$\mathrm{SD}(X(a,n), Y(a,n)) = \mathsf{neg}(n).$$

Throughout the paper, we use the notion of the associated matrices of a solitary output Boolean three-party functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$.

Definition 2.3 (Associated matrices of a 3-ary function)**.** *Fix a (possibly randomized) function* $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ *and let* $\mathcal{Z} = \{z_1, \ldots, z_k\}$*. We associate* k *matrices* $\mathbf{M}_{z_1}, \ldots, \mathbf{M}_{z_k} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{Y}|}$ *with the function* f *as follows. For every* $z \in \mathcal{Z}$*, the rows and columns of* $\mathbf{M}_z$ *are indexed with the elements of* $\mathcal{X}$ *and* $\mathcal{Y}$*, respectively, and each entry is defined as* $\mathbf{M}_z(x,y) = \Pr[f(x,y,z) = 1]$*, for all* $x \in \mathcal{X}$ *and* $y \in \mathcal{Y}$*. We refer to the collection* $\{\mathbf{M}_z\}_{z\in\mathcal{Z}}$ *as the* associated matrices *of* f*.*

2.2 The Model of Computation

We introduce the basic definitions for secure multi-party computation according to the real/ideal paradigm. We refer to [23] for more details. In short, a protocol is secure if whatever an adversary can do in the real world (i.e., a real execution of the protocol), can be done in the ideal world where the parties communicate with an uncorrupted trusted party that assists the computation. In this paper, we consider solitary output three-party functionalities. A *solitary output three-party functionality* is a sequence of a function $f = \{f_\kappa\}_{\kappa\in\mathbb{N}}$, where $f_\kappa : \mathcal{X}_\kappa \times \mathcal{Y}_\kappa \times \mathcal{Z}_\kappa \to \mathcal{W}_\kappa$ (for every value of the security parameter $\kappa \in \mathbb{N}$) is a random process that maps three-tuples of *inputs* to single random variable called *output*. We denote the parties by A, B, and C, where A holds $x \in \mathcal{X}$, B holds $y \in \mathcal{Y}$, and C holds $z \in \mathcal{Z}$. We let C be the party that receives the output. To alleviate notations, we remove κ from f and its domains and range and write it as $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{W}$.

The Real Model

Let A, B, and C denote the parties. A three-party protocol π is defined by a set of three PPT turning machines which are A, B, and C. Each party holds at the beginning of the execution the common security parameter 1^κ, a private input, and random coins. We denote the adversary by $\mathcal{A}$. The adversary is a PPT algorithm, which corrupts a subset of the parties and given an auxiliary input aux and the input of the corrupted parties. The adversary is static, that is, it chooses the subset it corrupts prior to the execution of the protocol. The adversary is assumed to be malicious, i.e., it

can instruct parties to deviate from the protocol in any arbitrary way during the execution, and is computationally bounded.

The parties execute the protocol over a synchronous network. That is, the execution proceeds in rounds, where each round consists of a *send phase* where parties send their messages for this round followed by a *receive phase*, where they receive messages from other parties. We consider a fully connected point-to-point network, where every pair of parties is connected by a communication line. We will further assume the parties have access to a broadcast channel. Throughout the execution of the protocol, all the honest parties follow the instructions of the prescribed protocol, whereas the corrupted parties receive their instructions from the adversary. The adversary has full access to the view of the corrupted parties, which consists of their inputs, their random coins, and the messages they see throughout this execution. At the end of the execution, the honest parties output their prescribed output from the protocol, the corrupted parties output nothing, and the adversary outputs a function of its view.

Finally, we let $\text{REAL}_{\pi,\mathcal{A}(\mathsf{aux})}(\kappa,(x,y,z))$ denote the view of the adversary and the output of the honest parties, in an execution of π on inputs (x,y,z) and security parameter κ, interacting with $\mathcal{A}$ with auxiliary input aux.

The Ideal Model

We consider an ideal computation with *guaranteed output delivery*, which is also referred to as *full security*, where a trusted party, denoted by T, performs the computation on behalf of the parties, and the ideal-model adversary cannot abort the computation. An ideal computation of a solitary output three-party functionality f, on inputs $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $z \in \mathcal{Z}$, and security parameter κ, with an ideal-world adversary $\mathcal{A}$ running with an auxiliary input aux and corrupting a subset $\mathcal{I} \subseteq \{\mathsf{A},\mathsf{B},\mathsf{C}\}$ of the parties, proceeds as follows:

Parties send inputs to the trusted party: Each honest party sends its input to T. For each corrupted party, the adversary $\mathcal{A}$ sends a value v from its domain as its input. Let (x',y',z') denote the inputs received by T.

Trusted party performs computation: If any $v \in \{x',y',z'\}$ is not in the appropriate domain (or was not sent at all), then T reassigns the aberrant input to some default value. Write (x',y',z') for the tuple of inputs after (possible) reassignment. The trusted party then chooses a random string r and computes $w = f(x',y',z';r)$.

Trusted party sends outputs: T sends w to C. If $\mathsf{C} \in \mathcal{I}$, then $\mathcal{A}$ receives w as well.

Outputs: If C is honest, then it outputs w. Otherwise, it outputs nothing. Both A and B output nothing, and the adversary $\mathcal{A}$ outputs a function of its view (i.e., the auxiliary input, its randomness, and the input and output of the corrupted parties).

We let $\text{IDEAL}_{\pi,\mathcal{A}(\mathsf{aux})}(\kappa,(x,y,z))$ denote the joint view of $\mathcal{A}$, being its output in a random execution of the above ideal-world process, and the output of the honest parties.

The Security Definition

Following the real vs. ideal paradigm, we next define security of protocols in the above model.

Definition 2.4 (Malicious security)**.** *Let f be a three-party functionality and let π be a three-party protocol. For a non-uniform adversary $\mathcal{A}$ corrupting controlling a subset $\mathcal{I} \subseteq \{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$, we say that π is* secure against *$\mathcal{A}$ if there exists a non-uniform polynomial time adversary $\mathcal{S}$ (called the simulator) controlling $\mathcal{I}$ in the ideal world such that*

$$\left\{\mathrm{REAL}_{\pi,\mathcal{A}(\mathsf{aux})}(\kappa,(x,y,z))\right\}_{\kappa\in\mathbb{N},x\in\mathcal{X},y\in\mathcal{Y},z\in\mathcal{Z},\mathsf{aux}\in\{0,1\}^*} \overset{\mathrm{C}}{\equiv} \left\{\mathrm{IDEAL}_{\pi,\mathcal{A}(\mathsf{aux})}(\kappa,(x,y,z))\right\}_{\kappa\in\mathbb{N},x\in\mathcal{X},y\in\mathcal{Y},z\in\mathcal{Z},\mathsf{aux}\in\{0,1\}^*}.$$

We say that π computes *f* with full security *if it is secure against any adversary corrupting a set of size at most 2.*

We now define the notion of backup values. In short, backup values are the values the honest parties use to recover their outputs when a corrupted party aborts after sending its previous messages honestly. In our case, only C learns the backup values. The backup values are well-defined for any fully secure protocol since such protocols must handle the case where a party aborts the computation.

Definition 2.5 (Backup values)**.** *Let f be a solitary-output three-party functionality, and let π be an r-round protocol computing f with full security. Sample the randomness of the parties, and consider an honest execution of π with the sampled randomness. For every $i \in \{0,\ldots,r\}$, the i^{th} backup value of (A,C), denoted a_i, is the output of an honest C in case party B aborted after sending i messages honestly (and party A remains honest). Similarly, the i^{th} backup value of (B,C), denoted b_i, is the output of an honest C in case party A aborted after sending i messages honestly.*

Although we are interested in full security, we will also use the weaker security notion called *security-with-identifiable-abort*.

Security-With-Identifiable-Abort

We now define an ideal computation with *security-with-identifiable-abort*. This model is somewhat similar to the ideal model where a trusted party performs the computation on behalf of the parties, but here the ideal-model adversary can abort the computation after learning the output, but with the cost of revealing the identity of a corrupted party. An ideal computation of a solitary output three-party functionality f, on inputs $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $z \in \mathcal{Z}$ and security parameter κ, with an ideal-world adversary $\mathcal{A}$ running with an auxiliary input aux and corrupting a subset $\mathcal{I} \subseteq \{\mathsf{A},\mathsf{B},\mathsf{C}\}$ of the parties, proceeds as follows:

Parties send inputs to the trusted party: Each honest party sends its input to T. For each corrupted party, the adversary $\mathcal{A}$ sends a value v from its domain as its input. Let (x',y',z') denote the inputs received by T.

Trusted party performs computation: If any $v \in \{x',y',z'\}$ is not in the appropriate domain (or was not sent at all), then T reassigns the aberrant input to some default value. Write (x',y',z') for the tuple of inputs after (possible) reassignment. The trusted party then chooses a random string r and computes $w = f(x',y',z';r)$.

Trusted party sends output to the adversary: If $\mathsf{C} \in \mathcal{I}$, then T sends w to $\mathcal{A}$.

Malicious adversary instructs the trusted party to continue or abort: The adversary $\mathcal{A}$ sends either $\mathsf{continue}$ or $(\mathsf{abort}, \mathsf{P})$ for some party $\mathsf{P} \in \mathcal{I}$. If it sends $\mathsf{continue}$, then T sends w to C if $\mathsf{C} \notin \mathcal{I}$. Otherwise, if $\mathcal{A}$ sends $(\mathsf{abort}, \mathsf{P})$, then T sends $(\mathsf{abort}, \mathsf{P})$ to the all honest parties.

Outputs: If C is honest, then it outputs whatever it received from the trusted party. Otherwise, it outputs nothing. Both A and B output whatever they got from T, and the adversary $\mathcal{A}$ outputs a function of its view (i.e., the auxiliary input, its randomness, and the input and output of the corrupted parties).

The Hybrid Model

The *hybrid model* is a model that combines both the real and ideal models. In this model, the parties can use an ideal trusted party to compute certain functionalities. The parties can communicate with this trusted party in the same way they can in the ideal model. Let f be a functionality. Then, an execution of a protocol π computing a functionality g in the f-hybrid model involves the parties sending normal messages to each other (as in the real model) and in addition, having access to a trusted party computing f. The invocations of f are done sequentially, that is before an invocation of f begins, the preceding invocation of f must finish. In particular, there is at most a single call to f per round, and no other messages are sent during any round in which f is called.

Let ρ be a protocol that securely computes f. The sequential composition theorem of Canetti [16] states that if a protocol π computes g in the f-hybrid model, then π^ρ securely computes g in the real model, where π^ρ is the protocol that is obtained by replacing all the ideal calls to the trusted party of π for computing f with the protocol ρ.

Proposition 2.6. *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{W}$ be a three-party functionality, let ρ be a protocol that computes f with security-with-identifiable-abort, and let π be a protocol that computes g with full security in the f-hybrid model. Then, π^ρ computes g with full security.*

Oblivious Transfer

We next define *1-out-of-2 oblivious transfer (OT)*. Intuitively, an OT protocol allows a sender holding two messages, m_0 and m_1, to send m_b to a receiver that holds the choice bit b. Security requires that the receiver learns only m_b (and nothing about m_{1-b}), and the sender learns nothing about b. Formally, the OT functionality $\mathrm{OT} : \{0,1\}^2 \times \{0,1\} \to \{0,1\}$ is defined as $\mathrm{OT}((m_0, m_1), b) = (\perp, m_b)$. A secure protocol for OT is one that computes OT with full security.

3 The Dealer Model for Solitary Output Three-Party Functionalities

Following previous similar works, it will be easier to describe our results in a *dealer model*. Here, the real world is augmented with a trusted dealer, which is a PPT algorithm that can interact with the parties in a limited way. Furthermore, the adversary is also limited when compared to a real-world adversary. Specifically, the adversary is assumed to be fail-stop, i.e., it acts honestly but may decide to abort prematurely. Additionally, it may change the input it sends to the dealer.

We formally describe the model below and show that it is equivalent to the real world (assuming OT), namely, any secure protocol in one model can be compiled into a secure protocol in the other model. The benefit of using this model is its simplicity in describing the protocols and the simpler security analysis. Additionally, our constructions achieve information-theoretic security. A similar model was already considered in the two-party setting [33, 8, 9], the multiparty setting [10, 12], and the security with friends and foes model [4].

We next describe a blueprint for an r-round protocol in the dealer model. The blueprint instructs the dealer to compute several backup values but does not specify how to compute these backup values. A protocol in the dealer model is obtained from the blueprint by defining the functions for computing these backup values. The interaction in the dealer model is as follows.

Definition 3.1 (Interaction in the dealer model)**.**
Private inputs: A *holds* $x \in \mathcal{X}$*,* B *holds* $y \in \mathcal{Y}$*, and* C *holds* $z \in \mathcal{Z}$*.*
Common inputs: All the parties hold the security parameter 1^κ*.*

1. *The honest parties send their inputs to the dealer, and the malicious adversary sends a value for every corrupted party. If the adversary does not send any input, the dealer replaces it with a default value.*

2. *The dealer first computes backup values* $a_0, \ldots, a_r$ *for* (A, C) *and* $b_0, \ldots, b_r$ *for* (B, C)*. It is required that* b_0 *and* a_0 *do not depend on the inputs of* A *and* B *respectively.*

3. *For every* $i \in \{0, \ldots, r\}$*, we let* $a_i[\mathsf{A}]$ *and* $a_i[\mathsf{C}]$ *denote the shares of* a_i*, and let* $b_i[\mathsf{B}]$ *and* $b_i[\mathsf{C}]$ *denote the shares of* b_i*. The dealer then shares each backup value in a 2-out-of-2 secret sharing scheme.*

4. *The dealer sends to* C *the shares* $(a_i[\mathsf{C}], b_i[\mathsf{C}])_{i=0}^{r}$*.*

5. *For* $i = 1$ *to* r*:*

   (a) *The dealer sends the share* $a_i[\mathsf{A}]$ *to party* A*.* A *then responds with either* continue *or* abort*.*

      i. *If* A *responds with* abort*, then the dealer approaches* B*, which responds with either* continue *or* abort*.*

      ii. *If* B *responds with* continue*, then the dealer sends* b_{i-1} *to* C *and halts. Party* C *outputs* b_{i-1}*.*

      iii. *If* B *responds with* abort*, then the dealer sends* $w = f(x_0, y_0, z)$ *for default values* $x_0 \in \mathcal{X}$ *and* $y_0 \in \mathcal{Y}$ *to* C *and halts. Party* C *outputs* w*.*

   (b) *The dealer sends the share* $b_i[\mathsf{B}]$ *to party* B*.* B *then responds with either* continue *or* abort*.*

      i. *If* B *responds with* abort*, then the dealer approaches* A*, which responds with either* continue *or* abort*.*

      ii. *If* A *responds with* continue*, then the dealer sends* a_i *to* C *and halts. Party* C *outputs* a_i*.*

      iii. *If* A *responds with* abort*, then the dealer sends* $w = f(x_0, y_0, z)$ *for default values* $x_0 \in \mathcal{X}$ *and* $y_0 \in \mathcal{Y}$ *to* C *and halts. Party* C *outputs* w*.*

6. *If no party aborted, the dealer approaches* A*, which responds with either* continue *or* abort*.*
   (a) *If* A *responds* continue*, then the dealer sends* a_r *to* C *and halts.* C *outputs* a_r*.*
   (b) *If* A *responds* abort*, then the dealer approaches to* B*, which responds with either* continue *or* abort*.*
   (c) *If* B *responds* continue*, then the dealer sends* b_r *to* C *and halts.* C *outputs* b_r*.*
   (d) *If no party responds with* continue*, then the dealer sends* $w = f(x_0, y_0, z)$ *for default values* $x_0 \in \mathcal{X}$ *and* $y_0 \in \mathcal{Y}$ *to* C *and halts.* C *outputs* w *in this case.*

We stress that the dealer is always honest in this model. It is also important to note that no adversary can force a corrupted C to deviate from the protocol, except for changing its input. Intuitively, this is because C is the only party to learn the output. Fix a protocol in the dealer model that is defined by the above definition. The security of the protocol is defined by comparing the execution in the dealer model to the ideal world defined previously. However, unlike the real world, here the malicious adversary is only fail-stop. Thus, we say the protocol computes f in the dealer model with full security if it computes f with full security against fail-stop adversaries. We next state the theorem that asserts that the dealer model is equivalent to the real model and prove it.

Theorem 3.2. *Let* $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{W}$ *be a solitary output three-party functionality. Then, assuming secure protocols for OT exist,* f *can be computed with full security in the real world if and only if it can be computed with full security in the dealer model.*

We prove the theorem by proving two lemmas, each handling a different direction of the statement.

Lemma 3.3. *Let* $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{W}$ *be a solitary output three-party functionality. Then, assuming secure protocols for OT exist and that* f *can be computed with full security in the dealer model. Then* f *can be computed with full security in the real world.*

Proof. Assume there is a protocol π^D computing f with full security in the dealer model against fail-stop adversaries. We construct a protocol π^R that computes f with full security in the real world. Fix a signature scheme $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ (since OT implies one-way functions [28] and one-way functions imply signature schemes [34], the assumption of the lemma implies signature schemes). Let ShrGen denote the three-party functionality roughly, given the parties' inputs, outputs a 3-out-of-3 secret sharing for each of the backup values computed by the dealer, each signed using the signature scheme. Formally, we define ShrGen as follows.

Functionality 3.4 (ShrGen)**.**
Private inputs: A *holds* $x \in \mathcal{X}$*,* B *holds* $y \in \mathcal{Y}$*, and* C *holds* $z \in \mathcal{Z}$*.*
Common inputs: All the parties hold the security parameter 1^κ*.*

1. *Sample signature scheme keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$
2. *For every* $i \in \{0, \ldots, r\}$*, compute the backup values* $(a_i)_{i=0}^r, (b_i)_{i=0}^r$ *as the dealer computes them.*
3. *Share the backup values* a_0 *and* b_0 *in a 2-out-of-2 secret sharing scheme. Then, compute the signatures*

- $\sigma_{a_0,\mathsf{A}} \leftarrow \mathsf{Sign}_{\mathsf{sk}}(a_0[\mathsf{A}])$
- $\sigma_{b_0,\mathsf{B}} \leftarrow \mathsf{Sign}_{\mathsf{sk}}(b_0[\mathsf{B}])$.

4. *For all $i \in \{1,\ldots,r\}$ share the backup values a_i and b_i in a 3-out-of-3 secret sharing scheme. Then, compute the signatures*

   - $\sigma_{a_i,\mathsf{A}} \leftarrow \mathsf{Sign}_{\mathsf{sk}}(a_i[\mathsf{A}])$
   - $\sigma_{b_i,\mathsf{A}} \leftarrow \mathsf{Sign}_{\mathsf{sk}}(b_i[\mathsf{A}])$
   - $\sigma_{a_i,\mathsf{B}} \leftarrow \mathsf{Sign}_{\mathsf{sk}}(a_i[\mathsf{B}])$
   - $\sigma_{b_i,\mathsf{B}} \leftarrow \mathsf{Sign}_{\mathsf{sk}}(b_i[\mathsf{B}])$.

5. *The parties obtain the following output.*

   - A *receives the public key* pk*, the shares of the backup values* $a_0[\mathsf{A}]$ *and* $(a_i[\mathsf{A}], b_i[\mathsf{A}])_{i=1}^r$*, and the signatures* $(\sigma_{a_i,\mathsf{A}}, \sigma_{b_i,\mathsf{A}})_{i=1}^r$ *and* $\sigma_{a_0,\mathsf{A}}$.
   - B *receives the public key* pk*, the shares of the backup values* $b_0[\mathsf{B}]$ *and* $(a_i[\mathsf{B}], b_i[\mathsf{B}])_{i=1}^r$*, and the signatures* $(\sigma_{a_i,\mathsf{B}}, \sigma_{b_i,\mathsf{B}})_{i=1}^r$ *and* $\sigma_{b_0,\mathsf{B}}$.
   - C *receives the public key* pk*, and the shares of the backup values* $(a_i[\mathsf{C}], b_i[\mathsf{C}])_{i=0}^r$.

We are now ready to describe the real-world protocol. For each party $\mathsf{P} \in \{\mathsf{A},\mathsf{B}\}$, we let $f_{-\mathsf{P}}$ denote the two-party solitary output functionality between the other two parties, obtained from f by fixing the input of P to a default value (x_0 if $\mathsf{P} = \mathsf{A}$ and y_0 if $\mathsf{P} = \mathsf{B}$). We consider the following three-party protocol π^R for computing f, described in the $\{\mathsf{ShrGen}, f_{-\mathsf{A}}, f_{-\mathsf{B}}\}$-hybrid model.

Protocol 3.5.
Private inputs: A *holds* $x \in \mathcal{X}$, B *holds* $y \in \mathcal{Y}$, *and* C *holds* $z \in \mathcal{Z}$.
Common inputs: All the parties hold the security parameter 1^κ.

1. *The parties call* ShrGen *with security-with-identifiable-abort, with their inputs.*
2. *If* C *aborts the execution, then the other parties halt.*
3. *If* $\mathsf{P} \in \{\mathsf{A},\mathsf{B}\}$ *aborts, then the remaining two parties call* $f_{-\mathsf{P}}$ *with their inputs and* C *outputs the result.*
4. *Otherwise, the parties do the following. For* $i = 1$ *to* r*:*
   (a) *Party* B *broadcasts* $(a_i[\mathsf{B}], \sigma_{a_i,\mathsf{B}})$.
   (b) *If* B *did not send any message or if* $\mathsf{Ver}_{\mathsf{pk}}(a_i[\mathsf{B}], \sigma_{a_i,\mathsf{B}}) = \mathsf{Fail}$*, then do the following.*
      i. A *sends* $(a_{i-1}[\mathsf{A}], \sigma_{a_{i-1},\mathsf{A}})$*, to* C.
      ii. *If* A *did not send any message or* $\mathsf{Ver}_{\mathsf{pk}}(a_{i-1}[\mathsf{A}], \sigma_{a_{i-1},\mathsf{A}}) = \mathsf{Fail}$*, then* C *outputs* $f(x_0, y_0, z)$ *for some default values* $x_0 \in \mathcal{X}$ *and* $y_0 \in \mathcal{Y}$.
      iii. *Otherwise,* C *outputs the backup value* a_{i-1}.
   (c) *Party* A *broadcasts* $(b_i[\mathsf{A}], \sigma_{b_i,\mathsf{A}})$.
   (d) *If* A *did not send any message or* $\mathsf{Ver}_{\mathsf{pk}}(b_i[\mathsf{A}], \sigma_{b_i,\mathsf{A}}) = \mathsf{Fail}$*, then do the following.*

i. B *sends* $(b_{i-1}[\mathsf{B}], \sigma_{b_{i-1},\mathsf{B}})$*, to* C*.*

ii. If B *did not send any message or* $\mathsf{Ver}_{\mathsf{pk}}(b_{i-1}[\mathsf{B}], \sigma_{b_{i-1},\mathsf{B}}) = \mathsf{Fail}$*, then* C *outputs* $f(x_0, y_0, z)$ *for some default values* $x_0 \in \mathcal{X}$ *and* $y_0 \in \mathcal{Y}$*.*

iii. Otherwise, C *outputs the backup value* b_{i-1}*.*

5. *If no abort occurred, then do the following.*

   (a) A *broadcasts* $(a_r[\mathsf{A}], \sigma_{a_r,\mathsf{A}})$*. If* $\mathsf{Ver}_{\mathsf{pk}}(a_r[\mathsf{A}], \sigma_{a_r,\mathsf{A}}) = \mathsf{Success}$*, then* C *outputs* $a_r = a_r[\mathsf{A}] + a_r[\mathsf{B}] + a_r[\mathsf{C}]$*.*

   (b) If A *did not send any message or* $\mathsf{Ver}_{\mathsf{pk}}(a_r[\mathsf{A}], \sigma_{a_r,\mathsf{A}}) = \mathsf{Fail}$*, then* B *broadcasts* $(b_r[\mathsf{B}], \sigma_{b_r,\mathsf{B}})$*. If* $\mathsf{Ver}_{\mathsf{pk}}(b_r[\mathsf{B}], \sigma_{b_r,\mathsf{B}}) = \mathsf{Success}$*, then* C *outputs* $b_r = b_r[\mathsf{A}] + b_r[\mathsf{B}] + b_r[\mathsf{C}]$*.*

   (c) If B *did not send any message or* $\mathsf{Ver}_{\mathsf{pk}}(b_r[\mathsf{B}], \sigma_{b_r,\mathsf{B}}) = \mathsf{Fail}$*, then* C *outputs* $f(x_0, y_0, z)$ *for some default values* $x_0 \in \mathcal{X}$ *and* $y_0 \in \mathcal{Y}$*.*

First, note that correctness is immediately implied by the correctness of the protocol in the dealer model stating that $a_r = f(x, y, z)$ except with negligible probability. We next show the security of the protocol. Observe that corrupting A or B (or both) will not provide the adversary any advantage since both A and B learn only random shares and the signatures corresponding to the shares. Thus, a simulator can generate random shares and signatures as the view of the adversary and send to the trusted party either default inputs or the inputs that were sent to ShrGen. Note that if C is the only corrupted party we can handle this case easily. This is due to the fact that the view of C until the last round consists only of random shares and signatures, and if it did not abort then also the output. Thus, a simulator can generate random shares and signatures as the view of C and send its input to the party depending if the adversary aborts or not.

We now focus on the case where both C and a party $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$ are corrupted. Note that in this case, the adversary can learn two backup values if C acts honestly and P aborts at some round. This could "help" the adversary to decide when to abort the protocol (if needed) based on the backup values it learned. We may only consider the case where $\mathsf{P} = \mathsf{A}$ since the case where $\mathsf{P} = \mathsf{B}$ is analogous. Let $\mathcal{A}^R$ be an adversary corrupting both A and C in the real world.

We separate the proof into two cases. For the first case, let us assume that $\mathcal{A}^R$ aborts a party (or two) during the call to ShrGen. If C aborts then the protocol halts, hence the view of $\mathcal{A}^R$ does not include any messages. On the other hand, if only A aborts during the call, then the view of $\mathcal{A}^R$ includes only the value $f_{-\mathsf{A}}(y, z) = f(x_0, y, z)$ (in addition to the corrupted parties' inputs). This can be simulated by sending (x_0, z) to T.

For the second case, we assume that $\mathcal{A}^R$ does not abort during the call to ShrGen. We construct an adversary $\mathcal{A}^D$ corrupting A and C in the dealer model that emulates $\mathcal{A}^R$. The adversary $\mathcal{A}^D$ works as follows. First, it samples signature scheme keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\mathsf{n}})$. Then, in each round $i \in [r]$, after the adversary $\mathcal{A}^D$ receives shares $(a_i[\mathsf{A}], a_i[\mathsf{C}])$ from the dealer, it computes the share $a_i[\mathsf{B}]$ as $a_i[\mathsf{B}] = a_i - (a_i[\mathsf{A}] + a_i[\mathsf{C}])$ and sign it. $\mathcal{A}^D$ sends $a_i[\mathsf{B}]$ and the signature to $\mathcal{A}^R$, which responds with its messages for A and C to the next round. If $\mathcal{A}^R$ instructs C (and possibly also A) to abort or to change its (signed) message, then $\mathcal{A}^D$ replies to the dealer with abort, outputs whatever $\mathcal{A}^R$ outputs and halts. If only A aborts or changes its message at some round i, then $\mathcal{A}^D$ sends abort to the dealer, and receives the backup value b_{i-1} from the dealer. It computes the message $b_{i-1}[\mathsf{B}] = b_{i-1} - (b_{i-1}[\mathsf{A}] + b_{i-1}[\mathsf{C}])$, sign it, send it to $\mathcal{A}^R$, output whatever it outputs, and halts.

Observe that the output of $\mathcal{A}^D$ is identically distributed to the output of $\mathcal{A}^R$. Hence, the simulator for $\mathcal{A}^D$ (assumed to exist), also simulates $\mathcal{A}^R$.

□

We now state and prove the second direction of the claim.

Lemma 3.6. *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{W}$ be a solitary output three-party functionality. Then, if f can be computed with full security in the real world, then f can be computed with full security in the dealer model.*

Proof. Assume there is a secure protocol π^R computing f in the real world. To describe a protocol in the dealer model, it suffices to describe the distribution of the backup values a_i and b_i for every $i \in \{0, \ldots, r\}$. The dealer computes these backup values by executing the protocol π^R in its head, and evaluating the backup values of each pair. That is, the dealer samples random coins for the parties and runs (an honest) execution of the protocol π^R in its head to compute the backup values a_i and b_i. To compute a_i it emulates the execution with the fixed random coins, assuming Party B aborts before sending the message in round i. Similarly, to compute b_i it emulates the execution with the fixed random coins, assuming Party A aborts before sending the message in round $i+1$.

After computing the backup values, the dealer proceeds with the computation, as described in Definition 3.1. Let π^D be the resulting protocol. Note that correctness is immediately implied by the fact that $a_r = f(x, y, z)$ except with negligible probability.

We prove that π^D is secure. Observe that corrupting either A or B or even both of them will not provide the adversary any advantages. This is true since in every round both A and B receive a random share of a_i and b_i, respectively. Thus, a simulator can simply sample these shares until the adversary aborts. Note that a corrupted C can be handled easily as well since C cannot abort during the computation in the dealer model (since the dealer never approaches C, but only sends to it its shares of a_i and b_i). Therefore, C's view can be simulated by sampling random shares as the view of C and send to T the same value it sends to the dealer.

We now focus on the case where both C and a party $\mathsf{P} \in \{\mathsf{A}, \mathsf{B}\}$ are corrupted. We consider only the case where $\mathsf{P} = \mathsf{A}$ since the case of $\mathsf{P} = \mathsf{B}$ is analogous. Let $\mathcal{A}^D$ be an adversary corrupting both A and C in the dealer model. Note that the adversary can instruct only A to abort. This is because by definition, the dealer never approaches C. We construct an adversary $\mathcal{A}^R$ corrupting A and C that emulates $\mathcal{A}^D$ in the real-world protocol π^R. First, it queries the dealer model adversary $\mathcal{A}^D$ to obtain the inputs it sends to the dealer. If $\mathcal{A}^D$ did not send an input for A or C, then $\mathcal{A}^R$ replaces the inputs of the appropriate parties with a default value. Then, $\mathcal{A}^R$ generates random shares $(a_i[\mathsf{C}], b_i[\mathsf{C}])_{i=0}^r$, sends them to $\mathcal{A}^D$ and computes the backup value a_0 and the share $a_0[\mathsf{A}] = a_0 - a_0[\mathsf{C}]$ (recall that we require that a_0 and b_0 to be independent the input of B). It then queries $\mathcal{A}^D$ with $a_0[\mathsf{A}]$ which replies with either $\mathsf{continue}$ or abort. If it responds with abort, then $\mathcal{A}^R$ instructs A to abort the computation and C to act honestly until the end of the computation. Otherwise, if $\mathcal{A}^D$ responds with $\mathsf{continue}$, then $\mathcal{A}^R$ instructs A to send the next message as an honest party would. Then, for $i \in \{1, \ldots, r\}$, $\mathcal{A}^R$ does the following.

1. Given the i^{th} message, $\mathcal{A}^R$ computes the backup value a_i and the share $a_i[\mathsf{A}] = a_i - a_i[\mathsf{C}]$.
2. $\mathcal{A}^R$ queries $\mathcal{A}^D$ with $a_i[\mathsf{A}]$ which replies with either $\mathsf{continue}$ or abort.
3. If $\mathcal{A}^D$ replies with abort, then $\mathcal{A}^R$ instructs A to abort the computation and C to act honestly until the end of the computation.

4. If $\mathcal{A}^D$ replies with continue, then $\mathcal{A}^R$ instructs A to send the next message as an honest party would.

Let b be the value that an honest C would have computed in the above computation. Observe that $\mathcal{A}^R$ can compute it because it does not instruct C to abort the computation. $\mathcal{A}^R$ queries $\mathcal{A}^D$ with b, outputs whatever $\mathcal{A}^D$ outputs, and halts. Note that like in the proof of Lemma 3.3, it is easy to see that the output of $\mathcal{A}^R$ is identically distributed to the output of $\mathcal{A}^D$. Hence, the simulator for $\mathcal{A}^R$ assumed to exist, also simulates $\mathcal{A}^D$.

□

3.1 Simple Security Properties of the Dealer Model

We now state and prove two claims asserting certain security properties of the dealer model. The first claim asserts that protocols in the dealer model that were constructed as described in Definition 3.1 are secure against any adversary corrupting only C, and the second claim reduces the security against an adversary corrupting both A and B to the case where exactly one of these parties is corrupted. For the rest of the section, we fix a solitary output Boolean three-party functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$.

Proposition 3.7. *Let π be a protocol in the dealer model following Definition 3.1 for computing f. Then, π is (perfectly) secure against any adversary corrupting only* C.

Proof. Let $\mathcal{A}$ be an adversary that corrupts C. We show that there exists an ideal world simulator that can generate a view for the ideal world adversary that is identically distributed to the view of $\mathcal{A}$ in the real world in an execution of π on f. The simulator is defined as follows

1. Query the adversary for the input z it sends to the dealer.
2. Sample random shares uniformly at random for C and denote them as $a_0[\mathsf{C}], \ldots, a_r[\mathsf{C}]$ and $b_0[\mathsf{C}], \ldots, b_r[\mathsf{C}]$.
3. Send z to the trusted party, receive $w = f(x, y, z)$ and set $a_r = w$.
4. Send the shares and w to $\mathcal{A}$, output whatever it outputs and halt.

Since only C is corrupted and in the dealer model the dealer never approaches it, the protocol never aborts. Therefore, its view in both worlds consists only of random shares and $f(x, y, z)$. □

Proposition 3.8. *Let π be a protocol in the dealer model following Definition 3.1 for computing f. Assume that π is secure against any adversary corrupting either* A *or* B. *Then π is secure against an adversary that corrupts both* A *and* B.

Proof. Let $\mathcal{A}$ be an adversary corrupting A and B. Before describing the simulator, we define two attackers, each corrupting a single party. The first adversary $\mathcal{A}_\mathsf{A}(y)$ is given an input $y \in \mathcal{Y}$ as auxiliary input. It corrupts A, samples the random shares for each backup value a_i and b_i, and instructs A to act the same as $\mathcal{A}$ instructs her on inputs x and y. The second adversary $\mathcal{A}_\mathsf{B}(x)$ is given x as an auxiliary input and is defined similarly. By the security assumption, there exist simulators $\mathcal{S}_\mathsf{A}(y)$ and $\mathcal{S}_\mathsf{B}(x)$ for $\mathcal{A}_\mathsf{A}(y)$ and $\mathcal{A}_\mathsf{B}(x)$, respectively. Observe that for every fixed choice of the randomness of the parties, every round $\mathcal{A}$ aborts A if and only if $\mathcal{A}_\mathsf{A}(y)$ aborts A, and every round $\mathcal{A}$ aborts B if and only if $\mathcal{A}_\mathsf{B}(x)$ aborts B.

We are now ready to define the simulator $\mathcal{S}$ for $\mathcal{A}$.

1. Query the adversary $\mathcal{A}$ for the inputs x and y it sends to the dealer.
2. Sample random shares for A and B and denote them as $a_0[\mathsf{A}], \ldots, a_r[\mathsf{A}]$ and $b_0[\mathsf{B}], \ldots, b_r[\mathsf{B}]$, respectively.
3. For $i = 0$ to r do the following:
   (a) Send the shares of a_i and b_i to $\mathcal{A}$.
   (b) If $\mathcal{A}$ instructs only A to abort at round i, then query $\mathcal{S}_{\mathcal{A}}(y)$ to obtain the input x^* it sends to T. Send x^* to T output whatever $\mathcal{A}$ outputs and halt.
   (c) If $\mathcal{A}$ instructs only B to abort at round i, then query $\mathcal{S}_{\mathcal{B}}(x)$ to obtain the input y^* it sends to T. Send y^* to T output whatever $\mathcal{A}$ outputs and halt.
   (d) If $\mathcal{A}$ instructs both A and B to abort at round i, then send to T the default values (x_0, y_0), output whatever $\mathcal{A}$ outputs and halt.
4. If no aborts occurred, then send (x, y) to T, output whatever $\mathcal{A}$ outputs and halt.

Observe that the view of the adversary in both of the worlds consists only of random shares, thus we only need to show that the output of C is identically distributed in both of the worlds. Note that if A and B never abort, then the output of C is $f(x, y, z)$ in both worlds. Additionally, if both A and B abort, then the output of C is $f(x_0, y_0, z)$ in both worlds. We may now assume that exactly one party aborts the computation.

By symmetry, we may only consider the case where A aborts the computation. Let $y \in \mathcal{Y}$ denote the input of B. In this case, $\mathcal{S}$ sends to the trusted party the same value that $\mathcal{S}_{\mathcal{A}}(y)$ sends. Therefore, the output of C is identically distributed in the two ideal worlds. By the security assumption against $\mathcal{A}_{\mathsf{A}}(y)$, it follows that the output of C in the ideal world when interacting with $\mathcal{S}_{\mathcal{A}}(y)$ is identically distributed to the output in the real world when interacting with $\mathcal{A}_{\mathsf{A}}(y)$. Since $\mathcal{A}_{\mathsf{A}}(y)$ aborts A if and only if $\mathcal{A}$ aborts A, it follows that the output of C in the real world when interacting with $\mathcal{A}_{\mathsf{A}}(y)$ is the identically distributed to the output when interacting with $\mathcal{A}$. We conclude that the output of C in the real world when interacting with $\mathcal{A}$ is identically distributed to the ideal world when interacting with $\mathcal{S}$. □

4 Positive Results: A General Family of Special-Round Protocols

In this chapter, we identify a class of solitary output functionalities that can be securely computed. Towards proving our result, we present a general family of protocols that follow the "special-round paradigm" [26, 9, 7], where the output is revealed at a special random round that is unknown to the parties. Looking ahead, in Section 5, we provide a new technique for analyzing when a given special-round protocol securely computes a given functionality. Furthermore, our analysis provides a general method for determining not only when a given protocol is secure, but it identifies *all* secure protocols (from the family) that securely compute a given functionality.

Before stating the theorem, let us define the notion of the projection matrices of a solitary output Boolean three-party functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$.

Definition 4.1 (Projection matrices of a 3-ary function)**.** *Fix a (possibly randomized) function $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$ and let $\mathcal{Z} = \{z_1, \ldots, z_k\}$. Recall that $\mathbf{M}_{z_1}, \ldots, \mathbf{M}_{z_k} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{Y}|}$ are the*

associated matrices of f. We denote by $\mathbf{M}_{\mathsf{c}}$ the matrix resulting by concatenating all the associated matrices by columns, that is,

$$\mathbf{M}_{\mathsf{c}} = (\mathbf{M}_{z_1} || \mathbf{M}_{z_2} || \dots || \mathbf{M}_{z_k}) ,$$

and we denote by $\mathbf{M}_{\mathsf{r}}$ the matrix resulting by concatenating all the associated matrices by rows, i.e.,

$$\mathbf{M}_{\mathsf{r}} = (\mathbf{M}_{z_1} ||_{\mathsf{r}} \mathbf{M}_{z_2} ||_{\mathsf{r}} \dots ||_{\mathsf{r}} \mathbf{M}_{z_k}) .$$

We refer to $\mathbf{M}_{\mathsf{r}}$ and $\mathbf{M}_{\mathsf{c}}$ as the projection matrices *of f.*[8]

Finally, for all $z \in \mathcal{Z}$, we let $\overline{\mathbf{M}}_z = \mathbf{J} - \mathbf{M}_z$, we let $\overline{\mathbf{M}}_{\mathsf{r}} = \mathbf{J} - \mathbf{M}_{\mathsf{r}}$, and we let $\overline{\mathbf{M}}_{\mathsf{c}} = \mathbf{J} - \mathbf{M}_{\mathsf{c}}$ be the complement matrices *of $\mathbf{M}_z$, $\mathbf{M}_{\mathsf{r}}$, and $\mathbf{M}_{\mathsf{c}}$, respectively.*

We prove the following theorem, identifying a class of Boolean three-party solitary output functionalities that can be securely computed. Roughly speaking, we show that if a certain system of linear equations and inequalities can be satisfied, then the functionality can be securely computed.

Theorem 4.2. *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a solitary output Boolean three-party functionality, and let $(\mathbf{M}_z)_{z \in \mathcal{Z}}$ be the associated matrices of f. Assume secure protocols for OT exist and that the following hold.*

1. *There exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{1} = 1$ and for every $z \in \mathcal{Z}$ it holds that $\mathbf{0}^T \leq \mathbf{p}^T \cdot \mathbf{M}_z \leq \mathbf{1}^T$.*
2. *For every $w \in \{0,1\}$ and every $z \in \mathcal{Z}$, there exists a matrix $\mathbf{X}_z^w \in \mathbb{R}^{|\mathcal{X}| \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ such that the following holds.*
   (a) *It holds that $\mathbf{0} \leq \mathbf{X}_z^w \cdot \mathbf{1} \leq \mathbf{1}$.*
   (b) *For every $x, x' \in \mathcal{X}$ and every $z' \in \mathcal{Z}$ where $z' \neq z$, it holds that $\mathbf{X}_z^w(x, (x', z')) \geq 0$.*
3. *It holds that $\mathbf{X}_z^1 \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{M}_z \cdot \mathbf{B}_z$, where $\mathbf{B}_z \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ is the diagonal matrix defined as $\mathbf{B}_z(y,y) = \mathbf{p}^T \cdot \mathbf{M}_z(\cdot, y)$ for all $y \in \mathcal{Y}$.*
4. *It holds that $\mathbf{X}_z^0 \cdot \overline{\mathbf{M}}_{\mathsf{r}} = \overline{\mathbf{M}}_z \cdot \overline{\mathbf{B}}_z$, where $\overline{\mathbf{B}}_z \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ is the diagonal matrix defined as $\overline{\mathbf{B}}_z(y,y) = \mathbf{p}^T \cdot \overline{\mathbf{M}}_z(\cdot, y) = 1 - \mathbf{B}(y,y)$ for all $y \in \mathcal{Y}$.*

Then f can be computed with full security.

The proof is given below. In the next section, we present a family of protocols and analyze when each of them is secure. The proof of the theorem is then followed by choosing an appropriate protocol.

4.1 A Family of Special-Round Protocols

In this section, we present a family of protocols that generalize the protocols that follow the "special-round paradigm" [26, 9, 7]. We then analyze their security. We call our family of protocols the *special-round protocols*. A special-round protocol is parametrized by a real value $0 < \alpha \leq 1$ and a vector $(\beta_{y,z})_{y \in \mathcal{Y}, z \in \mathcal{Z}} \in [0,1]^{|\mathcal{Y}| \cdot |\mathcal{Z}|}$. We assume that every party holds a private input (in addition

[8] The term projection comes from first viewing f as a 3-dimensional array, and then projecting it into two 2-dimensional matrices.

to the security parameter κ held as a common input). Specifically, A holds $x \in \mathcal{X}$, B holds $y \in \mathcal{Y}$, and C holds $z \in \mathcal{Z}$. In light of Theorem 3.2, we may define the protocols in the dealer model. Here it suffices to describe how the dealer computes the backup values of (A, C) and (B, C).

Before formally describing the protocols, we give an intuitive description. First, we denote the *geometric distribution with parameter* $\alpha > 0$ as $\mathrm{Geom}(\alpha)$, and it is defined as $\Pr_{i \leftarrow \mathrm{Geom}(\alpha)}[i = n] = (1-\alpha)^{n-1} \cdot \alpha$, for all integers $n \geq 1$. We further fix $r = r(\kappa) = \omega(\log \kappa)$ to be the number of rounds. We are now ready to describe the distribution of the backup values, given inputs x, y, and z of A, B, and C, respectively. The dealer samples $i^* \leftarrow \mathrm{Geom}(\alpha)$, then, before round i^* is reached, each pair among (A, C) and (B, C) together learn a random independent value, while after i^* is reached they learn the output. Finally, we define the backup value of (B, C) for round $i^* - 1$ to be 1 with probability $\beta_{y,z}$ (and 0 with probability $1 - \beta_{y,z}$). That is, if A aborts at round i^* then C outputs 1 with probability $\beta_{y,z}$.

We next formally describe the family of special-round protocols. Let $\alpha \in (0, 1]$ and $\boldsymbol{\beta} = (\beta_{y,z})_{y \in \mathcal{Y}, z \in \mathcal{Z}}$. We define the protocol $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ as follows.

Protocol 4.3 ($\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$)**.**
Private inputs: A *holds* $x \in \mathcal{X}$, B *holds* $y \in \mathcal{Y}$, *and* C *holds* $z \in \mathcal{Z}$.
Common inputs: All the parties hold the security parameter 1^κ.

1. *The dealer samples* $i^* \leftarrow \mathrm{Geom}(\alpha)$ *according to the geometric distribution with parameter* α.
2. *The dealer computes* $w = f(x, y, z)$. *It then sets* $\widetilde{b} = 1$ *with probability* $\beta_{y,z}$ *and* $\widetilde{b} = 0$ *with probability* $1 - \beta_{y,z}$.
3. *The dealer computes the backup values for every* $i \in \{0, \ldots, r\}$ *as*

$$a_i = \begin{cases} f(x, \widetilde{y}_i, z) & \text{if } i < i^* \\ w & \text{otherwise} \end{cases} \quad \text{and} \quad b_i = \begin{cases} f(\widetilde{x}_i, y, z) & \text{if } i < i^* \\ \widetilde{b} & \text{if } i = i^* - 1 \\ w & \text{otherwise} \end{cases}$$

   where $\widetilde{y}_i \leftarrow \mathcal{Y}$ *and* $\widetilde{x}_i \leftarrow \mathcal{X}$ *are all independent.*
4. *The dealer proceeds with the computation as described in Definition 3.1.*

The next lemma states sufficient conditions for $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ to be secure. Observe that it immediately implies Theorem 4.2.

Lemma 4.4. *Assume that the following holds for solitary output Boolean three-party functionality* $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0, 1\}$.

1. *There exists a vector* $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ *such that* $\mathbf{p}^T \cdot \mathbf{1} = 1$ *and for every* $z \in \mathcal{Z}$ *it holds that* $\mathbf{p}^T \cdot \mathbf{M}_z = (\beta_{y,z})^T_{y \in \mathcal{Y}}$.
2. *For every* $w \in \{0, 1\}$ *and every* $z \in \mathcal{Z}$, *there exists a matrix* $\mathbf{X}^w_z \in \mathbb{R}^{|\mathcal{X}| \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ *such that the following holds.*
   (a) *It holds that* $\mathbf{0} \leq \mathbf{X}^w_z \cdot \mathbf{1} \leq \mathbf{1}$.

(b) For every $x, x' \in \mathcal{X}$ and every $z' \in \mathcal{Z}$ where $z' \neq z$, it holds that $\mathbf{X}_z^w(x,(x',z')) \geq 0$.

3. *It holds that $\mathbf{X}_z^1 \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{M}_z \cdot \mathbf{B}_z$, where $\mathbf{B}_z \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ is the diagonal matrix defined as $\mathbf{B}_z(y,y) = \beta_{y,z}$ for all $y \in \mathcal{Y}$.*
4. *It holds that $\mathbf{X}_z^0 \cdot \overline{\mathbf{M}}_{\mathsf{r}} = \overline{\mathbf{M}}_z \cdot \overline{\mathbf{B}}_z$, where $\overline{\mathbf{B}}_z \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ is the diagonal matrix defined as $\overline{\mathbf{B}}_z(y,y) = 1 - \mathbf{B}(y,y) = 1 - \beta_{y,z}$ for all $y \in \mathcal{Y}$.*

Then, there exists α_0, such that for all $\alpha_0 \geq \alpha > 0$ it holds that $\pi_{\mathsf{sr}}^{\mathsf{so}}(\alpha,\boldsymbol{\beta})$ computes f with full security.

Proof. Observe that correctness holds since $i^* > r$ occurs with negligible probability. It remains to show that the protocol is fully secure. First, note that the case of a corrupted C is already handled by Proposition 3.7. The remaining cases are a corrupted A or B with possibly another corrupted party. Note that a corrupted B (and possibly another party) can be handled easily. Intuitively, this is due to the fact that A gets its share of a_i before B gets its share of b_i, thus A can help C learn the output first. We formally handle this case in Appendix A.2. The remaining cases are when the adversary corrupts only A, both A and C, and both A and B. Note that by Proposition 3.8, the last case is implied by the first case. We are left with the first two cases.

For a fixed $z \in \mathcal{Z}$ let $\boldsymbol{\beta}_z = (\beta_{y,z})_{y\in\mathcal{Y}}$. We next state and prove the two propositions that handle the first two cases. Interestingly, for the case where only A may be corrupted, we provide a characterization for when the protocol is secure against such an adversary.

Proposition 4.5. *$\pi_{\mathsf{sr}}^{\mathsf{so}}(\alpha,\boldsymbol{\beta})$ is secure against every adversary corrupting A for all sufficiently small $\alpha > 0$ if and only if there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{1} = 1$ and for every $z \in \mathcal{Z}$ it holds that $\mathbf{p}^T \cdot \mathbf{M}_z = \boldsymbol{\beta}_z^T$.*

Proof. Let $\mathbf{u}_{\mathcal{X}} \in \mathbb{R}^{|\mathcal{X}|}$ be the uniform probability vector over $\mathcal{X}$, i.e., $u_{\mathcal{X}}(x) = \frac{1}{|\mathcal{X}|}$ for every $x \in \mathcal{X}$. We first prove that $\pi_{\mathsf{sr}}^{\mathsf{so}}(\alpha,\boldsymbol{\beta})$ is secure against any adversary corrupting A if and only if for every $x \in \mathcal{X}$ and $i \in [r]$ there exists a probability vector $\mathbf{x}_{x,i} \in \mathbb{R}^{|\mathcal{X}|}$ such that for every $z \in \mathcal{Z}$ it holds that

$$\mathbf{x}_{x,i}^T \cdot \mathbf{M}_z = (1-\alpha)^i \cdot \mathbf{u}_{\mathcal{X}}^T \cdot \mathbf{M}_z + \alpha \cdot (1-\alpha)^{i-1} \cdot \boldsymbol{\beta}_z^T + (1-(1-\alpha)^{i-1}) \cdot \mathbf{e}_x^T \cdot \mathbf{M}_z. \tag{5}$$

We will then show that such a vector $\mathbf{x}_{x,i}$ exists if and only if there exists the vector $\mathbf{p}$ stated in the proposition.

First, observe that for any adversary corrupting A in the real world, its view consists of only random shares that are independent of the output of C. Since an ideal-world simulator can easily generate this view, security against the adversary holds if and only if there exists a distribution over $\mathcal{X}$ (which will be used by the simulator) such that the output distribution of C in the ideal world is identical to the real world. We show that Equation (5) encodes exactly that. That is, the y^{th} entry on the left-hand side is the output of C in an ideal-world execution when B holds input y, and the y^{th} entry on the right-hand side is the output of C in a real-world execution.

Fix a real-world adversary $\mathcal{A}$ that corrupts A. Let x be the input it sends to the dealer, and let i denote the round where it instructs A to abort (set to $r+1$ if no such round exists). Let $w(y)$ denote the output of C in the real-world when B holds input y. Then for all inputs $y \in \mathcal{Y}$ and

$z \in \mathcal{Z}$ it holds that

$$\begin{aligned}\Pr[w(y)=1] &= \Pr[i<i^*]\cdot\Pr[f(\widetilde{x}_i,y,z)=1] + \Pr[i=i^*]\cdot\beta_{y,z} + \Pr[i>i^*]\cdot\mathbf{M}_z(x,y)\\ &= (1-\alpha)^i\cdot\mathbf{u}_{\mathcal{X}}^T\cdot\mathbf{M}_z(\cdot,y) + \alpha\cdot(1-\alpha)^{i-1}\cdot\beta_{y,z}\\ &\quad + (1-(1-\alpha)^{i-1})\cdot\mathbf{e}_x^T\cdot\mathbf{M}_z(\cdot,y).\end{aligned}$$

Now, if there exists a probability vector $\mathbf{x}_{x,i}$ satisfying Equation (5), then the simulator for $\mathcal{A}$ will send to the trusted party a sample from $\mathbf{x}_{x,i}$. The formal simulator is defined as follows.

1. Query the adversary for the input x it sends to the dealer.
2. Sample $i^* \leftarrow \mathrm{Geom}(\alpha)$ according to the geometric distribution with parameter α.
3. For $i = 1$ to $i^* - 1$:
   (a) Send to $\mathcal{A}$ a random bit, which represents its share of the backup value a_i.
   (b) If $\mathcal{A}$ aborts A, then sample $x^* \leftarrow \mathbf{x}_{x,i}$, send x^* to the trusted party, and output random shares as the view of the adversary and halt.
4. Send x to the trusted party and receive $w = f(x,y,z)$.
5. For $i = i^*$ to r:
   - Send to $\mathcal{A}$ a random bit which represent its share from w.
   - If $\mathcal{A}$ aborts A then output random shares as the view of the adversary and halt.
6. output random shares as the view of the adversary and halt.

By Equation (5) security clearly holds. For the other direction, note that if there is a simulator for $\mathcal{A}$ then it must define such a probability vector.

To conclude the proof, we show that for every $x \in \mathcal{X}$ and $i \in [r]$ there exists a probability vector $\mathbf{x}_{x,i} \in \mathbb{R}^{|\mathcal{X}|}$ satisfying Equation (5) if and only if there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T\cdot\mathbf{M}_z = \boldsymbol{\beta}_z^T$ for all $z \in \mathcal{Z}$, and that $\mathbf{p}^T\cdot\mathbf{1} = 1$.

For the first direction, assume that there exists such a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$. Then

$$\begin{aligned}&(1-\alpha)^i\cdot\mathbf{u}_{\mathcal{X}}^T\cdot\mathbf{M}_z + \alpha\cdot(1-\alpha)^{i-1}\cdot\boldsymbol{\beta}_z^T + (1-(1-\alpha)^{i-1})\cdot\mathbf{e}_x^T\cdot\mathbf{M}_z\\ &\quad= (1-\alpha)^i\cdot\mathbf{u}_{\mathcal{X}}^T\cdot\mathbf{M}_z + \alpha\cdot(1-\alpha)^{i-1}\cdot\mathbf{p}^T\cdot\mathbf{M}_z + (1-(1-\alpha)^{i-1})\cdot\mathbf{e}_x^T\cdot\mathbf{M}_z\\ &\quad= \left((1-\alpha)^i\cdot\mathbf{u}_{\mathcal{X}}^T + \alpha\cdot(1-\alpha)^{i-1}\cdot\mathbf{p}^T + (1-(1-\alpha)^{i-1})\cdot\mathbf{e}_x^T\right)\cdot\mathbf{M}_z.\end{aligned}$$

Then $\mathbf{x}_{x,i} := (1-\alpha)^i\cdot\mathbf{u}_{\mathcal{X}} + \alpha\cdot(1-\alpha)^{i-1}\cdot\mathbf{p} + (1-(1-\alpha)^{i-1})\cdot\mathbf{e}_x$ satisfies Equation (5). It remains to show that it is also a probability vector. First, observe that

$$\begin{aligned}\mathbf{x}_{x,i}^T\cdot\mathbf{1} &= (1-\alpha)^i\cdot(\mathbf{u}_{\mathcal{X}}^T\cdot\mathbf{1}) + \alpha\cdot(1-\alpha)^{i-1}\cdot(\mathbf{p}^T\cdot\mathbf{1}) + (1-(1-\alpha)^{i-1})\cdot(\mathbf{e}_x^T\cdot\mathbf{1})\\ &= (1-\alpha)^i + \alpha\cdot(1-\alpha)^{i-1} + 1 - (1-\alpha)^{i-1}\\ &= 1.\end{aligned}$$

Now, observe that for every $x' \in \mathcal{X}$ and all sufficiently small $\alpha > 0$ it holds that

$$\begin{aligned} x_{x,i}(x') &\geq (1-\alpha)^i \cdot \frac{1}{|\mathcal{X}|} + \alpha \cdot (1-\alpha)^{i-1} \cdot p(x') \\ &\geq 0. \end{aligned}$$

Note that the second term tends to 0 as α tends to 0. Thus, $x_{x,i}(x')$ is arbitrarily close to $(1-\alpha)^i \cdot \frac{1}{|\mathcal{X}|} \geq 0$.

For the second direction, we assume that the vector $\mathbf{x}_{x,i}$ exists for all $i \in [r]$. Letting $i = 1$ and isolating $\boldsymbol{\beta}_z^T$ results in

$$\frac{1}{\alpha} \cdot \left(\mathbf{x}_{x,i}^T - (1-\alpha) \cdot \mathbf{u}_{\mathcal{X}}^T\right) \cdot \mathbf{M}_z = \boldsymbol{\beta}_z^T.$$

Therefore, the vector $\mathbf{p}^T = \frac{1}{\alpha} \cdot \left(\mathbf{x}_{x,i}^T - (1-\alpha) \cdot \mathbf{u}_{\mathcal{X}}^T\right)$ is mapped to $\boldsymbol{\beta}_z^T$ for every $z \in \mathcal{Z}$. It remains to show that $\mathbf{p}^T \cdot \mathbf{1} = 1$. Indeed, since $\mathbf{x}_{x,i}$ and $\mathbf{u}_{\mathcal{X}}$ are both probability vectors, it follows that

$$\begin{aligned} \mathbf{p}^T \cdot \mathbf{1} &= \frac{1}{\alpha} \cdot \left(\mathbf{x}_{x,i}^T - (1-\alpha) \cdot \mathbf{u}_{\mathcal{X}}^T\right) \cdot \mathbf{1} \\ &= \frac{1}{\alpha} \cdot \left(\mathbf{x}_{x,i}^T \cdot \mathbf{1} - (1-\alpha) \cdot \mathbf{u}_{\mathcal{X}}^T \cdot \mathbf{1}\right) \\ &= \frac{1}{\alpha} \cdot (1 - (1-\alpha)) \\ &= 1. \end{aligned}$$

□

Proposition 4.6. *Assume that for every $z \in \mathcal{Z}$ there exist matrices $\mathbf{X}_z^1$, $\mathbf{X}_z^0 \in \mathbb{R}^{|\mathcal{X}| \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ for which Items 2 to 4 of Lemma 4.4 hold. Then for all sufficiently small $\alpha > 0$, $\pi_{\mathsf{sr}}^{\mathsf{so}}(\alpha, \boldsymbol{\beta})$ is secure against any adversary corrupting both A and C.*

Proof. Fix a real-world adversary $\mathcal{A}$ that corrupts both A and C. First, recall that the adversary cannot abort C in the dealer model. The simulator works as follows.

1. Query the adversary for the inputs x and z it sends to the dealer.
2. Sample random shares $(a_i[\mathsf{C}], b_i[\mathsf{C}])_{i=0}^r$ and send them to $\mathcal{A}$.
3. Sample $i^* \leftarrow \mathrm{Geom}(\alpha)$ according to the geometric distribution with parameter α.
4. For $i = 1$ to $i^* - 1$:
   (a) Compute $a_i = f(x, \widetilde{y}_i, z)$, where $\widetilde{y}_i \leftarrow \mathcal{Y}$, and send $a_i[\mathsf{A}] := a_i - a_i[\mathsf{C}]$ to $\mathcal{A}$.
   (b) If $\mathcal{A}$ aborts A then do the following.
      i. Sample $(x^*, z^*) \leftarrow \mathbf{v}_{x,z}^{a_i}$, where $\mathbf{v}_{x,z}^{a_i}$ is a probability vector to be defined by the analysis below.
      ii. Send (x^*, z^*) to the trusted party and receive $w = f(x^*, y, z^*)$.
      iii. Send $b_{i-1}[\mathsf{B}] := w - b_{i-1}[\mathsf{C}]$ to the adversary, output whatever it outputs, and halt.
5. Send (x, z) to the trusted party and receive $w = f(x, y, z)$.

6. Send $a_{i^*}[\mathsf{A}] = w - a_{i^*}[\mathsf{C}]$ to $\mathcal{A}$.

   - If the adversary aborts A at round i^* then do the following.
     (a) Set $b_{i^*-1} = 1$ with probability $\gamma^w_{x,z}$ and $b_{i^*-1} = 0$ with probability $1 - \gamma^w_{x,z}$.
     (b) Send $b_{i^*-1}[\mathsf{B}] := b_{i^*-1} - b_{i^*-1}[\mathsf{C}]$ to the adversary, output whatever it outputs, and halt.

7. For $i = i^* + 1$ to r:
   (a) Send $a_i[\mathsf{A}] = w - a_i[\mathsf{C}]$ to $\mathcal{A}$.
   (b) If $\mathcal{A}$ aborts A then send it $b_{i-1}[\mathsf{B}] := w - b_{i-1}[\mathsf{C}]$, output whatever it outputs, and halt.

8. If $\mathcal{A}$ aborts A then send it $b_r[\mathsf{B}] := w - b_r[\mathsf{C}]$, output whatever it outputs, and halt.

9. Otherwise, output whatever $\mathcal{A}$ outputs and halt.

Let i denote the round where the adversary instructs A to abort (set to $r+1$ if no such round exists, or if the adversary aborts at the end). Observe that in the real world, the adversary holds enough shares to reconstruct the backup values $a_1, \ldots, a_i$ and b_{i-1}. Note that if $i > i^*$, then the view of the adversary in both worlds is equal to $a_1, \ldots, a_i$ and b_{i-1}, where $a_i = b_{i-1} = f(x, y, z)$. Therefore, we may condition on the event $i \leq i^*$. Also, note that given $i \leq i^*$, all the backup values $a_1, \ldots, a_{i-1}$ and b_{i-1} are independent in both worlds. Thus, it is enough to analyze the distribution of (a_i, b_{i-1}) in both worlds. In the following, we fix the inputs $x \in \mathcal{X}$ and $z \in \mathcal{Z}$ that $\mathcal{A}$ sends to the dealer.

We first introduce some notations. Let $s_{x,z} = \Pr_{\widetilde{y} \leftarrow \mathcal{Y}}[f(x, \widetilde{y}, z) = 1]$ and let $\mathbf{u}_z \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{Z}|}$ be the vector defined as

$$u_z(x', z') = \begin{cases} \frac{1}{|\mathcal{X}|} & \text{if } z' = z \\ 0 & \text{otherwise} \end{cases}$$

for all $x' \in \mathcal{X}$ and $z' \in \mathcal{Z}$.

Let us first analyze the probability that $(a_i, b_{i-1}) = (1, 1)$. In the real world, it holds that

$$\begin{aligned} \Pr[(a_i, b_{i-1}) = (1,1) \mid i \leq i^*] &= (1-\alpha) \cdot \Pr[f(x, \widetilde{y}_i, z) = 1] \cdot \Pr[f(\widetilde{x}_i, y, z) = 1] \\ &\quad + \alpha \cdot \mathbf{M}_z(x, y) \cdot \beta_{y,z} \\ &= (1-\alpha) \cdot s_{x,z} \cdot \mathbf{u}_z^T \cdot \mathbf{M}_{\mathsf{r}}(\cdot, y) + \alpha \cdot \mathbf{M}_z(x, y) \cdot \beta_{y,z}. \end{aligned}$$

On the other hand, in the ideal world, it holds that

$$\begin{aligned} \Pr[(a_i, b_{i-1}) = (1,1) \mid i \leq i^*] &= (1-\alpha) \cdot \Pr[f(x, \widetilde{y}_i, z) = 1] \cdot \Pr[f(x^*, y, z^*) = 1 \mid a_i = 1] \\ &\quad + \alpha \cdot \mathbf{M}_{\mathsf{r}}((x, z), y) \cdot \gamma^1_{x,z} \\ &= (1-\alpha) \cdot s_{x,z} \cdot \mathbf{v}^1_{x,z} \cdot \mathbf{M}_{\mathsf{r}}(\cdot, y) + \alpha \cdot \mathbf{e}^T_{x,z} \cdot \mathbf{M}_{\mathsf{r}}(\cdot, y) \cdot \gamma^1_{x,z}, \end{aligned}$$

where $\mathbf{e}_{x,z} \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{Z}|}$ is the $(x, z)^{\text{th}}$ standard basis vector.

For the protocol to be secure, it must hold that the distribution of (a_i, b_{i-1}) in the real world is identical to the distribution of (a_i, b_{i-1}) in the ideal world. Thus, it must hold that

$$\begin{aligned}&(1-\alpha)\cdot s_{x,z}\cdot \mathbf{u}_z^T\cdot \mathbf{M}_{\mathsf{r}}(\cdot,y)+\alpha\cdot \mathbf{M}_z(x,y)\cdot \beta_{y,z}\\&\quad=(1-\alpha)\cdot s_{x,z}\cdot \mathbf{v}_{x,z}^1\cdot \mathbf{M}_{\mathsf{r}}(\cdot,y)+\alpha\cdot \mathbf{e}_{x,z}^T\cdot \mathbf{M}_{\mathsf{r}}(\cdot,y)\cdot \gamma_{x,z}^1.\end{aligned}\tag{6}$$

Observe that if $s_{x,z}=0$ then it means that $f(x,\cdot,z)\equiv 0$ is the constant 0 function. Therefore, both sides of Equation (6) are 0. Assume now that $s_{x,z}\neq 0$. Then, isolating $\mathbf{v}_{x,z}^1$ results in

$$\mathbf{v}_{x,z}^1\cdot \mathbf{M}_{\mathsf{r}}(\cdot,y)=\mathbf{u}_z^T\cdot \mathbf{M}_{\mathsf{r}}(\cdot,y)+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \mathbf{M}_z(x,y)\cdot \beta_{y,z}-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \mathbf{e}_{x,z}^T\cdot \mathbf{M}_{\mathsf{r}}(\cdot,y)\cdot \gamma_{x,z}^1,$$

Since this must hold for all $y\in\mathcal{Y}$, we get that

$$\mathbf{v}_{x,z}^1\cdot \mathbf{M}_{\mathsf{r}}=\mathbf{u}_z^T\cdot \mathbf{M}_{\mathsf{r}}+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot (\mathbf{M}_z(x,y)\cdot \beta_{y,z})_y^T-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1\cdot \mathbf{e}_{x,z}^T\cdot \mathbf{M}_{\mathsf{r}}.$$

Now, recall that by Item 3 of Lemma 4.4, it holds that $\mathbf{X}_z^1(x,\cdot)\cdot \mathbf{M}_{\mathsf{r}}=(\mathbf{M}_z(x,y)\cdot \beta_{y,z})_y^T$. Therefore $\mathbf{v}_{x,z}^1$ must satisfy

$$\mathbf{v}_{x,z}^1\cdot \mathbf{M}_{\mathsf{r}}=\left(\mathbf{u}_z^T+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \mathbf{X}_z^1(x,\cdot)-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1\cdot \mathbf{e}_{x,z}^T\right)\cdot \mathbf{M}_{\mathsf{r}}.$$

Setting $\mathbf{v}_{x,z}^1=\mathbf{u}_z^T+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \mathbf{X}_z^1(x,\cdot)-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1\cdot \mathbf{e}_{x,z}^T$ solves the equation. It remains to show that it is also a probability vector. First, since $\mathbf{u}_z$ and $\mathbf{e}_{x,z}^T$ are probability vectors and since $\mathbf{X}_z^1(x,\cdot)\cdot \mathbf{1}=\gamma_{x,z}^1$, it holds that

$$\begin{aligned}\mathbf{v}_{x,z}^1\cdot \mathbf{1}&=\mathbf{u}_z^T\cdot \mathbf{1}+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot (\mathbf{X}_z^1(x,\cdot)\cdot \mathbf{1})-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1\cdot \mathbf{e}_{x,z}^T\cdot \mathbf{1}\\&=1+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1\\&=1.\end{aligned}$$

Next, we show that $v_{x,z}^1(x',z')\geq 0$ for all $x'\in\mathcal{X}$ and $z'\in\mathcal{Z}$. We consider two cases, depending on whether $z'=z$ or not. For the first case where $z'\neq z$, it holds that $u_z(x,z')=e_{x,z}(x,z')=0$, hence

$$v_{x,z}^1(x',z')=\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \mathbf{X}_z^1(x',z')\geq 0,$$

where the inequality is by Item 2b of Lemma 4.4. Next, for $z'=z$ it holds that

$$\begin{aligned}v_{x,z}^1(x',z)&\geq\frac{1}{|\mathcal{X}|}+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \mathbf{X}_z^1(x',z)-\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \gamma_{x,z}^1\\&=\frac{1}{|\mathcal{X}|}+\frac{\alpha}{(1-\alpha)\cdot s_{x,z}}\cdot \left(\mathbf{X}_z^1(x',z)-\gamma_{x,z}^1\right).\end{aligned}$$

Note that the second term tends to 0 as α tends to 0. Thus, $v_{x,z}^1(x',z)$ is arbitrarily close to $\frac{1}{|\mathcal{X}|}\geq 0$, hence it is positive for all sufficiently small $\alpha>0$. We conclude that $\mathbf{v}_{x,z}^1$ is a probability vector.

It remains to consider the remaining three cases, i.e., the probability that $(a_i, b_{i-1}) = (0,0)$, that $(a_i, b_{i-1}) = (1,0)$, and that $(a_i, b_{i-1}) = (0,1)$. The case of $(a_i, b_{i-1}) = (0,0)$ is analogous to the case of $(a_i, b_{i-1}) = (1,1)$, and is implied by Items 2 and 4. Next, we argue that the case of $(a_i, b_{i-1}) = (0,1)$ is equivalent to the case of $(a_i, b_{i-1}) = (0,0)$ and the case of $(a_i, b_{i-1}) = (1,0)$ is equivalent to the case of $(a_i, b_{i-1}) = (1,1)$. To see this, first observe that

$$\Pr[a_i = 0] = \Pr[a_i = 0, b_{i-1} = 0] + \Pr[a_i = 0, b_{i-1} = 1].$$

Since a_i is identically distributed in both worlds and we proved that the probability that $(a_i, b_{i-1}) = (0,0)$ is the same in both worlds as well, it follows that the probability that $(a_i, b_{i-1}) = (0,1)$ is the same. Since all probabilities sum to 1, it follows that the probability that $(a_i, b_{i-1}) = (1,0)$ is also the same as the probability that $(a_i, b_{i-1}) = (1,1)$. □

□

Example 4.7. *We next give an example of functionality that can be securely computed using* $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \beta)$. *First, let* $\mathcal{D} = \{\mathcal{S} \subseteq [3] : 1 \leq |\mathcal{S}| \leq 2\}$, *and define the functions* $\mathsf{disj} : \mathcal{D} \times \mathcal{D} \to \{0,1\}$ *and* $\mathsf{size}_{\mathsf{eq}} : \mathcal{D} \times \mathcal{D} \to \{0,1\}$ *as*

$$\mathsf{disj}(\mathcal{S}, \mathcal{T}) = \begin{cases} 1 & \text{if } \mathcal{S} \cap \mathcal{T} = \emptyset \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mathsf{size}_{\mathsf{eq}}(\mathcal{S}, \mathcal{T}) = \begin{cases} 1 & \text{if } |\mathcal{S}| = |\mathcal{T}| \\ 0 & \text{otherwise} \end{cases}.$$

Consider the solitary output Boolean three-party functionality $f : \mathcal{D} \times \mathcal{D} \times \{0,1\} \to \{0,1\}$ *defined as*

$$f(\mathcal{S}, \mathcal{T}, z) = \begin{cases} \mathsf{disj}(\mathcal{S}, \mathcal{T}) & \text{if } z = 0 \\ \mathsf{size}_{\mathsf{eq}}(\mathcal{S}, \mathcal{T}) & \text{if } z = 1 \end{cases}.$$

We use Theorem 4.2 to show that f *can be securely computed. For that, consider the associated matrices of* f, *given by*

$$\mathbf{M}_0 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{M}_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Let $\mathbf{p} = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})^T$ *be the uniform probability vector of dimension 6. Then,*

$$\mathbf{p}^T \cdot \mathbf{1} = 1, \quad \mathbf{p}^T \cdot \mathbf{M}_0 = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)^T, \quad \text{and} \quad \mathbf{p}^T \cdot \mathbf{M}_1 = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)^T.$$

Thus, Item 1 of Theorem 4.2 holds. It remains to show that there exist matrices $\mathbf{X}^1_0, \mathbf{X}^0_0, \mathbf{X}^1_1, \mathbf{X}^0_1 \in \mathbb{R}^{6 \times 12}$ *such that*

$$\mathbf{X}^1_0 \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{M}_0 \cdot \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{6} \end{pmatrix}, \quad \mathbf{X}^0_0 \cdot \overline{\mathbf{M}}_{\mathsf{r}} = \overline{\mathbf{M}}_0 \cdot \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{5}{6} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{5}{6} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{5}{6} \end{pmatrix},$$

$$\mathbf{X}_1^1 \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{M}_1 \cdot \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}, \quad \mathbf{X}_1^0 \cdot \overline{\mathbf{M}}_{\mathsf{r}} = \overline{\mathbf{M}}_1 \cdot \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix},$$

and for all $w, z \in \{0,1\}$ *and* $x, x' \in \mathcal{D}$ *it holds that*

$$\mathbf{0} \leq \mathbf{X}_z^w \cdot \mathbf{1} \leq \mathbf{1} \quad \text{and} \quad 0 \leq \mathbf{X}_z^w(x,(x',1-z)).$$

Indeed, taking

$$\mathbf{X}_0^1 = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{X}_0^0 = \begin{pmatrix} \frac{1}{2} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & 0 & 0 & 0 \\ -\frac{1}{3} & \frac{1}{2} & -\frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & 0 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & 0 & 0 & 0 \end{pmatrix},$$

and

$$\mathbf{X}_1^1 = \mathbf{X}_1^0 = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

satisfy the requirements.

4.2 Solving the Equations: A Necessary Condition for Security

In this section, we show that for $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ to compute f with full security, then either Items 3 and 4 of Lemma 4.4 must hold, or f can be computed "trivially." Here, trivial means that a secure protocol can be obtained by switching the roles of A and B and taking $\boldsymbol{\beta} = \mathbf{0}$. Note that by Lemma 4.4, this results in a secure protocol if $\mathbf{0}$ is an affine combination of the columns of $\mathbf{M}_{\mathsf{r}}$. We next formally state and prove the result. Interestingly, unlike our other results, this only holds for functionalities that are independent of κ. Additionally, we need to assume that α is constant.

Lemma 4.8. *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a solitary output Boolean three-party functionality, let $\alpha > 0$ be a constant, and let $\boldsymbol{\beta} = (\beta_{y,z})_{y \in \mathcal{Y}, z \in \mathcal{Z}} \in [0,1]^{|\mathcal{Y}| \cdot |\mathcal{Z}|}$. Assume that there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{Z}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{1}^T$, and that $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ computes f with full security. Then for every $z \in \mathcal{Z}$, there exists a matrix $\mathbf{X} \in \mathbb{R}^{(|\mathcal{X}| \cdot |\mathcal{Z}|) \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ for which it holds that*

$$\mathbf{X} \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{M}_z \cdot \mathbf{B}_z,$$

where $\mathbf{B}_z \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ is the diagonal matrix for which the values on the main diagonal are $\beta_{y,z}$ for all $y \in \mathcal{Y}$.

Toward proving Lemma 4.8, we use the following proposition proved by [9].

Proposition 4.9 ([9, Proposition 2.1])**.** *For any matrix $\mathbf{M}$, it holds that $\mathbf{0}^T$ is an affine combination of the rows of $\mathbf{M}$ if and only if $\mathbf{1}$ is* not *a linear combination of the columns of $\mathbf{M}$.*

Proof of Lemma 4.8. For the remaining of the proof, we fix $x \in \mathcal{X}$ and $z \in \mathcal{Z}$. We first describe two adversaries for $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$. For $w \in \{0,1\}$, let $\mathcal{A}_w$ be the adversary that corrupts both A and C and works as follows.

1. Instruct A and C to send x and z, respectively, to the dealer and act honestly in the first round of the execution.
2. Reconstruct the backup value a_1 using the shares given by the dealer.
3. If $a_1 = w$, then instruct A to abort the execution. The dealer then sends b_0 to C. Output (a_1, b_0) and halt.
4. Otherwise, if $a_1 \neq w$ then instruct A to act honestly until the end of the protocol. The dealer then sends a_r to C. Output (a_1, a_r) and halt.

Denote by out_w the second coordinate in the output of $\mathcal{A}_w$. By the security assumption, there exists a simulator Sim_w (that may depend on κ) that simulates $\mathcal{A}_w$ in the ideal world. Let $D_{w,\kappa}$ denote the distribution over $\mathcal{X} \times \mathcal{Z}$ that is used to sample the inputs x^* and z^* sent by Sim_w to the trusted party. Denote by S_w denote the algorithm that on input $(\kappa, x, z, x^*, z^*, f(x^*, y, z^*))$, runs Sim_w given that it sent x^* and z^* to the trusted party, and outputs whatever it outputs.[9] Then assuming $(x^*, z^*) \leftarrow D_{w,\kappa}$, it follows that output distribution of $\mathsf{S}_w(\kappa, x, z, x^*, z^*, f(x^*, y, z^*))$ is identical to the output distribution of the simulator. By the security assumption, it follows that

$$\{\mathsf{S}_w(\kappa, x, z, x^*, z^*, f(x^*, y, z^*))\}_{\kappa \in \mathbb{N}, y \in \mathcal{Y}} \overset{\mathrm{C}}{\equiv} \left\{\mathrm{REAL}_{\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha,\boldsymbol{\beta}),\mathcal{A}_w}(\kappa, x, y, z)\right\}_{\kappa \in \mathbb{N}, y \in \mathcal{Y}}.$$

[9] Note that S_w might be inefficient (i.e., its running time is not necessarily polynomial in the security parameter). We note that this will not affect the correctness of the proof since we care about existence.

Since the domain of f is of constant size, it follows that the ensembles are statistically close. In particular, it holds that

$$\left|\Pr\left[\mathsf{S}_w(\kappa, x, z, x^*, z^*, f(x^*, y, z^*)) = (1,1)\right] - \Pr\left[(a_1, \mathsf{out}_w) = (1,1)\right]\right| = \mathsf{neg}(\kappa). \tag{7}$$

We now analyze the two probabilities. We do so only for $w = 1$, which shows that $(\mathbf{M}_z(x,y) \cdot \beta_{y,z})_{y \in \mathcal{Y}} \in \mathrm{Im}(\mathbf{M}_{\mathsf{r}}^T)$. The other case is similar and proves the second part of the lemma. For brevity, we remove w from all notations.

We start by analyzing the second term, which corresponds to the real world. Let $s = \Pr_{\widetilde{y} \leftarrow \mathcal{Y}}\left[f(x, \widetilde{y}, z) = 1\right]$ and let $\mathbf{u} \in \mathbb{R}^{|\mathcal{X}| \cdot |\mathcal{Z}|}$ be the vector defined as

$$u(x', z') = \begin{cases} \frac{1}{|\mathcal{X}|} & \text{if } z' = z \\ 0 & \text{otherwise} \end{cases}$$

for all $x' \in \mathcal{X}$ and $z' \in \mathcal{Z}$. For every $y \in \mathcal{Y}$ it holds that

$$\begin{aligned} \Pr\left[(a_1, \mathsf{out}) = (1,1)\right] &= \Pr\left[1 < i^*\right] \cdot \Pr_{\widetilde{y} \leftarrow \mathcal{Y}}\left[f(x, \widetilde{y}, z) = 1\right] \cdot \Pr_{\widetilde{x} \leftarrow \mathcal{X}}\left[f(\widetilde{x}, y, z) = 1\right] \qquad (8) \\ &\quad + \Pr\left[i^* = 1\right] \cdot \mathbf{M}_z(x,y) \cdot \beta_{y,z} \\ &= (1-\alpha) \cdot s \cdot \mathbf{u}^T \cdot \mathbf{M}_{\mathsf{r}}(\cdot, y) + \alpha \cdot \mathbf{M}_z(x,y) \cdot \beta_{y,z}. \qquad (9) \end{aligned}$$

We next analyze the probability that S outputs $(1,1)$. We first introduce some notations. Let $\mathbf{d}_\kappa \in [0,1]^{|\mathcal{X}| \cdot |\mathcal{Z}|}$ denote the probability vector that corresponds to the distribution D_κ. For a possible output $\mathsf{out}' \in \{0,1\}$, let

$$q_{\kappa, x^*, z^*, \mathsf{out}'} = \Pr_{(x^*, z^*) \leftarrow D_\kappa}\left[\mathsf{S}(\kappa, x, z, x^*, z^*, \mathsf{out}') = (1,1)\right].$$

and let $\mathbf{Q}_{\kappa, \mathsf{out}'} \in [0,1]^{(|\mathcal{X}| \cdot |\mathcal{Z}|) \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ be the diagonal matrix whose diagonals are defined as $\mathbf{Q}_{\kappa, \mathsf{out}'}((x^*, z^*), (x^*, z^*)) = q_{\kappa, x^*, z^*, \mathsf{out}'}$ for all $x^* \in \mathcal{X}$ and $z^* \in \mathcal{Z}$. Finally, let $\mathbf{P} \in \mathbb{R}^{(|\mathcal{X}| \cdot |\mathcal{Z}|) \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ be the matrix where all its rows are the vector $\mathbf{p}^T$, and let $\mathbf{I} \in \mathbb{R}^{(|\mathcal{X}| \cdot |\mathcal{Z}|) \times (|\mathcal{X}| \cdot |\mathcal{Z}|)}$ be the identity matrix.

Observe that

$$\begin{aligned} &\Pr_{(x^*, z^*) \leftarrow D_\kappa}\left[\mathsf{S}(\kappa, x, z, x^*, z^*, f(x^*, y, z^*)) = (1,1)\right] \\ &= \sum_{\substack{x^* \in \mathcal{X}, \\ z^* \in \mathcal{Z}}} d_\kappa(x^*, z^*) \cdot \left(\mathbf{M}_{\mathsf{r}}((x^*, z^*), y) \cdot q_{\kappa, x^*, z^*, 1} + \overline{\mathbf{M}}_{\mathsf{r}}((x^*, z^*), y) \cdot q_{\kappa, x^*, z^*, 0}\right) \\ &= \mathbf{d}_\kappa^T \cdot \left(\mathbf{Q}_{\kappa,1} \cdot \mathbf{M}_{\mathsf{r}} + \mathbf{Q}_{\kappa,0} \cdot \overline{\mathbf{M}}_{\mathsf{r}}\right) \cdot \mathbf{e}_y \\ &= \mathbf{d}_\kappa^T \cdot \left(\mathbf{Q}_{\kappa,1} \cdot \mathbf{M}_{\mathsf{r}} + \mathbf{Q}_{\kappa,0} \cdot (\mathbf{J} - \mathbf{M}_{\mathsf{r}})\right) \cdot \mathbf{e}_y \\ &= \mathbf{d}_\kappa^T \cdot \left(\mathbf{Q}_{\kappa,1} \cdot \mathbf{M}_{\mathsf{r}} + \mathbf{Q}_{\kappa,0} \cdot (\mathbf{P} \cdot \mathbf{M}_{\mathsf{r}} - \mathbf{M}_{\mathsf{r}})\right) \cdot \mathbf{e}_y \\ &= \mathbf{d}_\kappa^T \cdot \left(\mathbf{Q}_{\kappa,1} + \mathbf{Q}_{\kappa,0} \cdot (\mathbf{P} - \mathbf{I})\right) \cdot \mathbf{M}_{\mathsf{r}} \cdot \mathbf{e}_y. \end{aligned}$$

Combining this with Equations (7) and (8), it follows that

$$\left|(1-\alpha) \cdot s \cdot \mathbf{u}^T \cdot \mathbf{M}_{\mathsf{r}} \cdot \mathbf{e}_y + \alpha \cdot \mathbf{M}_z(x,y) \cdot \beta_{y,z} - \mathbf{d}_\kappa^T \cdot \left(\mathbf{Q}_{\kappa,1} + \mathbf{Q}_{\kappa,0} \cdot (\mathbf{P} - \mathbf{I})\right) \cdot \mathbf{M}_{\mathsf{r}} \cdot \mathbf{e}_y\right| \leq \mathsf{neg}(\kappa),$$

for all $y \in \mathcal{Y}$. Therefore, the vector

$$\frac{1}{\alpha} \cdot \Big((1-\alpha) \cdot s \cdot \mathbf{u}^T - \mathbf{d}_\kappa^T \cdot (\mathbf{Q}_{\kappa,1} + \mathbf{Q}_{\kappa,0} \cdot (\mathbf{P} - \mathbf{I})) \Big) \cdot \mathbf{M}_\mathsf{r}$$

tends to $(\mathbf{M}_z(x,y) \cdot \beta_{y,z})_{y \in \mathcal{Y}}$ as $\kappa \to \infty$. Recall that α is constant and all entries in the two vectors are probabilities, hence the preimage belongs to $[-1/\alpha, 1/\alpha]^{|\mathcal{X}| \cdot |\mathcal{Z}|}$. Since this set is mapped by $\mathbf{M}_\mathsf{r}$ to a closed set (in the topological sense), there exists $\mathbf{v}$ such that $\mathbf{v} \cdot \mathbf{M}_\mathsf{r} = (\mathbf{M}_z(x,y) \cdot \beta_{y,z})_{y \in \mathcal{Y}}$. Since it must hold for all $x \in \mathcal{X}$, there exists $\mathbf{X}$ for which it holds that $\mathbf{X} \cdot \mathbf{M}_\mathsf{r} = \mathbf{M}_z \cdot \mathbf{B}_z$. □

5 A Complete Analysis of the System of Equations

In this chapter, we abstract the system of equations that arose from the analysis of Protocol 4.3. Let $n, m, k \in \mathbb{N}$ and fix two collections of matrices $\mathcal{N} = \{\mathbf{N}_z \in \mathbb{R}^{n \times \ell}\}_{z \in [k]}$ and $\mathcal{M} = \{\mathbf{M}_z \in \mathbb{R}^{m \times \ell}\}_{z \in [k]}$, and a vector $\boldsymbol{\beta} \in \mathbb{R}^{k \cdot \ell}$ whose coordinates are indexed with a pair $(y,z) \in [\ell] \times [k]$. The system for $\mathcal{N}$, $\mathcal{M}$, and $\boldsymbol{\beta}$, denoted $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$, is defined as follows. For every $z \in [k]$ let $\mathbf{B}_z \in \mathbb{R}^{k \times k}$ be the diagonal matrix whose y^{th} entry on the diagonal is $\beta_{y,z}$, and let $\boldsymbol{\beta}_z = (\beta_{y,z})_{y \in [\ell]}$. Then the system is given by

$$\begin{cases} \forall z \in [k] : \ \mathbf{X}_z \cdot \mathbf{N}_z = \mathbf{M}_z \cdot \mathbf{B}_z \\ \forall z \in [k] : \ \mathbf{p}^T \cdot \mathbf{M}_z = \boldsymbol{\beta}_z^T \\ \mathbf{p}^T \cdot \mathbf{1} = 1 \end{cases} \tag{10}$$

where the unknowns are the matrices $\mathbf{X}_z \in \mathbb{R}^{m \times n}$ and the vector $\mathbf{p} \in \mathbb{R}^n$. We call $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$ solvable if there exist $(\mathbf{X}_z)_{z \in [k]}$ and $\mathbf{p}$ that satisfy Equation (10).

As an example, recall that in the way we used the system in Section 4, the collection of matrices $\mathcal{N}$ and $\mathcal{M}$ were defined by the functionality to be computed, the vector $\boldsymbol{\beta}$ defines the special-round protocol to be used to compute the functionality, and $(\mathbf{X}_z)_{z \in [k]}$ and $\mathbf{p}$ translate to the simulators for the adversaries. That is, given a solitary output three-party functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$, the family $\mathcal{N}$ contains only the projection matrix $\mathbf{M}_\mathsf{r}$ (i.e., $\mathbf{N}_z = \mathbf{M}_\mathsf{r}$ for all $z \in [k]$), and the family $\mathcal{M}$ is the set of all associated matrices of f. See Example 5.4 below for a concrete example.

Since we are interested in when a secure protocol exists, in this section we consider the following question:

Given $\mathcal{N}$ and $\mathcal{M}$, for what values of $\boldsymbol{\beta}$ is $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$ solvable?

Note that an answer does not necessarily provide us with a secure three-party protocol for solitary output functionalities since the security also requires additional inequalities to hold. However, by Proposition 4.5 and Lemma 4.8 a solution to the system is a necessary condition for security. Therefore, we obtain an equivalent simpler formulation for the necessary condition (see Corollary 5.3 below for a formal statement). Additionally, this allows us to find an example of a functionality where no special-round protocol computes it with full security, and our impossibility result from Section 6 does not apply (see Example 5.4).

Interestingly, our analysis can be applied to other settings. We showcase this in Section 5.2, where we show how our analysis can be applied to the setting of fair two-party computation for Boolean functionalities. Similarly to the solitary output setting, here we are also able to provide necessary and sufficient conditions for the security of special-round protocols, though here the gap

is much smaller. This provides stronger results than what was shown by [26, 8, 31, 9]. Thus, our analysis could be of independent interest and may find use in other models and more general settings as well.

5.1 Characterizing the Existence of Solutions for the Equations

In this section, we characterize for what $\boldsymbol{\beta}$ the system $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$ is solvable. Roughly, we show that a solution exists if and only if the image of a certain matrix contains the first standard basis vector $\mathbf{e}_1$. Before formally stating our results, we first introduce some notations.

Notations. For a matrix $\mathbf{M}$ we denote its rank by $\mathsf{rank}(\mathbf{M})$. We further denote by $\mathsf{rref}(\mathbf{M})$ its *reduced row echelon form* and by $\mathsf{rref}^*(\mathbf{M})$ its reduced row echelon form with the rows of zeros removed. Given a matrix $\mathbf{R} \in \mathbb{R}^{m \times \ell}$ that is in reduced row echelon form, a column $y \in [\ell]$ is called a pivot if $\mathbf{R}(\cdot, y)$ is a standard basis vector and for all $y' < y$ (according to lexicographic ordering) $\mathbf{R}(\cdot, y') \neq \mathbf{R}(\cdot, y)$. The column is called free otherwise. We let $\mathcal{P}_{\mathbf{R}}$ denote the set of pivots in $\mathbf{R}$, and let $\mathcal{F}_{\mathbf{R}}$ denote the set of free columns in $\mathbf{R}$. Finally, we denote the *pivot-to-row* function of $\mathbf{R}$ by $\mathsf{ptr}_{\mathbf{R}} : \mathcal{P}_{\mathbf{R}} \to [m]$. That is, it is the function that given a pivot $y \in \mathcal{P}_{\mathbf{R}}$, returns the unique row $x \in [m]$ such that $\mathbf{R}(x, y) = 1$. We prove the following.

Theorem 5.1. *Let $n, m, k \in \mathbb{N}$ and fix two collections of matrices $\mathcal{N} = \{\mathbf{N}_z \in \mathbb{R}^{n \times \ell}\}_{z \in [k]}$ and $\mathcal{M} = \{\mathbf{M}_z \in \mathbb{R}^{m \times \ell}\}_{z \in [k]}$. For every $z \in [k]$, denote $\mathbf{R}_z = \mathsf{rref}^*(\mathbf{N}_z)$, and define the matrix $\mathbf{K}_z \in \mathbb{R}^{(m \cdot (\ell - \mathsf{rank}(\mathbf{N}_z))) \times \ell}$ as*

$$\mathbf{K}_z((x, y_{\mathsf{free}}), y) = \begin{cases} \mathbf{M}_z(x, y) \cdot \mathbf{R}_z(\mathsf{ptr}_{\mathbf{R}_z}(y), y_{\mathsf{free}}) & \text{if } y \in \mathcal{P}_{\mathbf{R}_z} \\ -\mathbf{M}_z(x, y) & \text{if } y = y_{\mathsf{free}} \\ 0 & \text{otherwise} \end{cases}$$

for all $x \in [m]$, $y_{\mathsf{free}} \in \mathcal{F}_{\mathbf{R}_z}$, and $y \in [\ell]$. Finally, let

$$\mathbf{L} = \left(\mathbf{1} ||_{\mathsf{r}} \left((\mathbf{M}_1 || \dots || \mathbf{M}_k) \cdot (\mathbf{K}_1 || \dots || \mathbf{K}_k)^T \right)\right).$$

There exists a vector $\boldsymbol{\beta} \in \mathbb{R}^{k \cdot \ell}$ such that $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$ is solvable if and only if $\mathbf{e}_1 \in \mathrm{Im}(\mathbf{L})$, where $\mathbf{e}_1$ is the first standard basis vector. Moreover, if $\mathbf{L} \cdot \mathbf{v} = \mathbf{e}_1$ then $\boldsymbol{\beta} = (\mathbf{M}_1 || \dots || \mathbf{M}_k)^T \cdot \mathbf{v}$ is such that $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$ is solvable.

To prove the Theorem 5.1, we first characterize when a matrix $\mathbf{X}_z$ exists for a single z. Roughly, we show that a solution exists if and only if the entries in the vector $\boldsymbol{\beta}$ satisfy some linear relation.

Lemma 5.2. *Let $\mathbf{N} \in \mathbb{R}^{n \times \ell}$, let $\mathbf{M} \in \mathbb{R}^{m \times \ell}$, and let $\boldsymbol{\beta} \in \mathbb{R}^{\ell}$. Denote by $\mathbf{B} \in \mathbb{R}^{\ell \times \ell}$ the diagonal matrix whose y^{th} entry is β_y. Finally, let $\mathbf{R} = \mathsf{rref}^*(\mathbf{N})$ and define the matrix $\mathbf{K} \in \mathbb{R}^{(m \cdot (\ell - \mathsf{rank}(\mathbf{N}))) \times \ell}$ as*

$$\mathbf{K}((x, y_{\mathsf{free}}), y) = \begin{cases} \mathbf{M}(x, y) \cdot \mathbf{R}(\mathsf{ptr}_{\mathbf{R}}(y), y_{\mathsf{free}}) & \text{if } y \in \mathcal{P}_{\mathbf{R}} \\ -\mathbf{M}(x, y) & \text{if } y = y_{\mathsf{free}} \\ 0 & \text{otherwise} \end{cases}$$

for all $x \in [m]$, $y_{\mathsf{free}} \in \mathcal{F}_{\mathbf{R}}$, and $y \in [\ell]$. Then there exist a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ such that $\mathbf{X} \cdot \mathbf{N} = \mathbf{M} \cdot \mathbf{B}$ if and only if $\boldsymbol{\beta} \in \mathrm{Ker}(\mathbf{K})$.

Proof. We first transform the system into an equivalent system. Let $\widehat{\mathbf{R}} \in \mathbb{R}^{n \times \ell}$ denote the reduced row echelon form of $\mathbf{N}$ and let $\mathbf{E} \in \mathbb{R}^{n \times n}$ be the matrix that transforms $\mathbf{N}$ to $\widehat{\mathbf{R}}$ using elementary row operations. Then $\mathbf{E} \cdot \mathbf{N} = \widehat{\mathbf{R}}$. Since row reduction is an invertible process, $\mathbf{E}$ has an inverse $\mathbf{E}^{-1} \in \mathbb{R}^{n \times n}$. Let $\widehat{\mathbf{X}} = \mathbf{X} \cdot \mathbf{E}^{-1}$. Then

$$\widehat{\mathbf{X}} \cdot \widehat{\mathbf{R}} = \mathbf{X} \cdot \mathbf{E}^{-1} \cdot \mathbf{E} \cdot \mathbf{N} = \mathbf{X} \cdot \mathbf{N} = \mathbf{M} \cdot \mathbf{B}.$$

Let $n' = \mathsf{rank}(\mathbf{N})$. Recall that $\mathbf{R} \in \mathbb{R}^{n' \times \ell}$ is the matrix $\widehat{\mathbf{R}}$ with the rows of zeros removed. Then a solution $\widehat{\mathbf{X}}$ exists if and only if there exists $\widetilde{\mathbf{X}} \in \mathbb{R}^{m \times n'}$ such that

$$\widetilde{\mathbf{X}} \cdot \mathbf{R} = \mathbf{M} \cdot \mathbf{B}. \tag{11}$$

We may now only concern ourselves with analyzing Equation (11).

To analyze Equation (11), we first show that a solution for the set of columns in $\mathcal{P}_{\mathbf{R}}$ fixes the solution matrix $\widetilde{\mathbf{X}}$. We then use this to analyze the system for the columns in $\mathcal{F}_{\mathbf{R}}$. Observe that for every $y_{\mathsf{piv}} \in \mathcal{P}_{\mathbf{R}}$ it holds that

$$\beta_{y_{\mathsf{piv}}} \cdot \mathbf{M}(\cdot, y_{\mathsf{piv}}) = \mathbf{M} \cdot \mathbf{B}(\cdot, y_{\mathsf{piv}}) = \widetilde{\mathbf{X}} \cdot \mathbf{R}(\cdot, y_{\mathsf{piv}}) = \widetilde{\mathbf{X}}(\cdot, \mathsf{ptr}_{\mathbf{R}}(y_{\mathsf{piv}})).$$

Then for every $y_{\mathsf{free}} \in \mathcal{F}_{\mathbf{R}}$ it follows that

$$\beta_{y_{\mathsf{free}}} \cdot \mathbf{M}(\cdot, y_{\mathsf{free}}) = \mathbf{M} \cdot \mathbf{B}(\cdot, y_{\mathsf{free}}) = \widetilde{\mathbf{X}} \cdot \mathbf{R}(\cdot, y_{\mathsf{free}}) = \sum_{x \in [n']} \widetilde{\mathbf{X}}(\cdot, x) \cdot \mathbf{R}(x, y_{\mathsf{free}}).$$

Since $\mathsf{ptr}_{\mathbf{R}} : \mathcal{P}_{\mathbf{R}} \to [n']$ is bijective, it follows that

$$\begin{aligned}
\beta_{y_{\mathsf{free}}} \cdot \mathbf{M}(\cdot, y_{\mathsf{free}}) &= \sum_{x \in [n']} \widetilde{\mathbf{X}}(\cdot, x) \cdot \mathbf{R}(x, y_{\mathsf{free}}) \\
&= \sum_{y_{\mathsf{piv}} \in \mathcal{P}_{\mathbf{R}}} \widetilde{\mathbf{X}}(\cdot, \mathsf{ptr}_{\mathbf{R}}(y_{\mathsf{piv}})) \cdot \mathbf{R}(\mathsf{ptr}_{\mathbf{R}}(y_{\mathsf{piv}}), y_{\mathsf{free}}) \\
&= \sum_{y_{\mathsf{piv}} \in \mathcal{P}_{\mathbf{R}}} \beta_{y_{\mathsf{piv}}} \cdot \mathbf{M}(\cdot, y_{\mathsf{piv}}) \cdot \mathbf{R}(\mathsf{ptr}_{\mathbf{R}}(y_{\mathsf{piv}}), y_{\mathsf{free}}).
\end{aligned}$$

We conclude that if there exists a solution to Equation (11), then it must hold that $\mathbf{K} \cdot \boldsymbol{\beta} = \mathbf{0}$. Moreover, if $\mathbf{K} \cdot \boldsymbol{\beta} = \mathbf{0}$, then the above analysis shows that $\widetilde{\mathbf{X}}(\cdot, \mathsf{ptr}_{\mathbf{R}}(y_{\mathsf{piv}})) = \beta_{y_{\mathsf{piv}}} \cdot \mathbf{N}(\cdot, y_{\mathsf{piv}})$ for all $y_{\mathsf{piv}} \in \mathcal{Y}_{\mathsf{piv}}$ is a solution to Equation (11). □

We are now ready to prove Theorem 5.1.

Proof of Theorem 5.1. By Lemma 5.2, the matrices $(\mathbf{X}_z)_{z \in [k]}$ exist if and only if $(\mathbf{K}_1 || \ldots || \mathbf{K}_k) \cdot \boldsymbol{\beta} = \mathbf{0}$. Therefore, for $\mathbf{p}$ to exist it must satisfy

$$(\mathbf{K}_1 || \ldots || \mathbf{K}_k) \cdot (\mathbf{M}_1 || \ldots || \mathbf{M}_k)^T \cdot \mathbf{p} = \mathbf{0} \quad \text{and} \quad \mathbf{1}^T \cdot \mathbf{p} = 1.$$

Since this is equivalent to $\mathbf{L} \cdot \mathbf{p} = \mathbf{e}_1$, the theorem follows. □

Combining Theorem 5.1, Proposition 4.5, and Lemma 4.8, provides a simple necessary condition for the existence of a special-round protocol computing a solitary output three-party Boolean functionality.

Corollary 5.3. *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a solitary output three-party Boolean functionality, let $\mathcal{N} = \{\mathbf{M}_{\mathsf{r}}\}$ (i.e., $\mathbf{N}_z = \mathbf{M}_{\mathsf{r}}$ for all $z \in \mathcal{Z}$), let $\mathcal{M} = \{\mathbf{M}_z\}_{z\in\mathcal{Z}}$, and define $\mathbf{L}$ as in Theorem 5.1. Assume that there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|\cdot|\mathcal{Z}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_{\mathsf{r}} = \mathbf{1}^T$. If $\mathbf{e}_1 \notin \mathbf{L}$ then no special-round protocol $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ with a constant $\alpha > 0$ computes f with full security.*

Example 5.4. *Consider the following randomized solitary output functionality $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$, whose associated matrices $\mathbf{M}_0$ and $\mathbf{M}_1$ are defined by the following matrices.*

$$\mathbf{M}_0 = \begin{pmatrix} 0 & 1/4 & 1/2 \\ 1/4 & 1/2 & 3/4 \\ 1/2 & 3/4 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{M}_1 = \begin{pmatrix} 0 & 1/3 & 2/3 \\ 0 & 1/3 & 2/3 \\ 0 & 1/3 & 2/3 \end{pmatrix},$$

where $\mathcal{X} = \{x_1, x_2, x_3\}$, $\mathcal{Y} = \{y_1, y_2, y_3\}$, and $\mathcal{Z} = \{0,1\}$. First, note that $\mathbf{M}_0$ and $\mathbf{M}_1$ describe two 2-ary functionalities, f_0 which is associated with $\mathbf{M}_0$ and f_1 which is associated with $\mathbf{M}_1$. The functionality f describes two 3-ary extensions. The first is of f_0, that is $f(\cdot,\cdot,0) \equiv f_0(\cdot,\cdot,\lambda)$, and the second is of f_1, that is $f(\cdot,\cdot,1) \equiv f_1(\cdot,\cdot,\lambda)$. We now show that no special-round protocol $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ with a constant $\alpha > 0$ can compute f with full security. Note that since f is not strong semi-balanced, we cannot apply Theorem 6.3 to show it is impossible to compute. Thus, f is an example of a function whose status is unknown.

The reduced row echelon form matrices (without the rows of zeros) $\mathbf{R}_0$ and $\mathbf{R}_1$ of $\mathbf{M}_0$ and $\mathbf{M}_1$, respectively, are given by

$$\mathbf{R}_0 = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{pmatrix} \quad \text{and} \quad \mathbf{R}_1 = \begin{pmatrix} 0 & 1 & 2 \end{pmatrix}.$$

Then the matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ are defined as

$$\mathbf{K}_0 = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{4} & 1 & -\frac{3}{4} \\ -\frac{1}{2} & \frac{3}{2} & -1 \end{pmatrix} \quad \text{and} \quad \mathbf{K}_1 = \begin{pmatrix} 0 & \frac{2}{3} & -\frac{2}{3} \\ 0 & \frac{2}{3} & -\frac{2}{3} \\ 0 & \frac{2}{3} & -\frac{2}{3} \end{pmatrix}.$$

Recall that $\mathbf{L} = \left(\mathbf{1}||_{\mathsf{r}} \left((\mathbf{M}_0||\mathbf{M}_1) \cdot (\mathbf{K}_0||\mathbf{K}_1)^T\right)\right)$. Then,

$$\mathbf{L} = \begin{pmatrix} 1 & 1 & 1 \\ -\frac{25}{72} & -\frac{25}{72} & -\frac{25}{72} \\ -\frac{25}{72} & -\frac{25}{72} & -\frac{25}{72} \\ -\frac{25}{72} & -\frac{25}{72} & -\frac{25}{72} \end{pmatrix}.$$

Since $\mathbf{e}_1 \notin \mathrm{Im}(\mathbf{L})$, by Corollary 5.3 it follows that no special-round protocol $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ with a constant $\alpha > 0$ computes f with full security.

We stress that the above functionality cannot be computed using $\pi^{\mathsf{so}}_{\mathsf{sr}}(\alpha, \boldsymbol{\beta})$ for any constant $\alpha > 0$ and vector $\boldsymbol{\beta}$, but we do not know if this functionality can be computed with full security in general.

5.2 Applications to Two-Party Fair Computation

We next show how the above results can be applied to the two-party setting for Boolean functionalities. Recall that Asharov et al. [9] characterizes the set of symmetric Boolean functionalities (i.e., both parties output the same bit as the output) that can be computed with full security in this setting. Our results, however, strengthen [9], as we present a family of special-round protocols and analyze (using Theorem 5.1) for a given functionality which of the protocols securely computes it. We believe our techniques could be used to construct secure protocols for several more settings, e.g., *non-Boolean* asymmetric functionalities. Thus, this could also allow us to improve the techniques of [21, 32] for constructing fair protocols, which used locking strategies to determine which special-round protocols one should use.

We next present the family of special-round protocols for the two-party setting. Similarly to the solitary output setting, the family of protocols we construct here naturally generalize previous constructions [26, 8, 9]. We further describe them in the dealer model for the two-party setting, which [8, 9] showed how to compile them to a protocol in the real world. In short, the dealer is defined similarly to the solitary output setting, but instead of sending A and B their shares of a_i and b_i, respectively, the dealer sends them the actual value of a_i and b_i. We next formalize the description of the protocols. Let $\alpha \in (0,1]$ and let $\boldsymbol{\beta} \in [0,1]^{|\mathcal{Y}|}$. We define the protocol $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ as follows.

Protocol 5.5 ($\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$)**.**
Private inputs: A *holds* $x \in \mathcal{X}$ *and* B *holds* $y \in \mathcal{Y}$.
Common inputs: Both parties hold the security parameter 1^κ.

1. A *and* B *send their private inputs* x *and* y *to the dealer.*
2. *The dealer samples* $i^* \leftarrow \mathrm{Geom}(\alpha)$ *according to the geometric distribution with parameter* α.
3. *The dealer computes* $w = f(x,y)$. *It then sets* $\widetilde{b} = 1$ *with probability* β_y *and sets* $\widetilde{b} = 0$ *with probability* $1 - \beta_y$.
4. *The dealer computes the backup values for every* $i \in \{0, \ldots, r\}$ *as*

$$a_i = \begin{cases} f(x, \widetilde{y}_i) & \text{if } i < i^* \\ w & \text{otherwise} \end{cases} \quad \text{and} \quad b_i = \begin{cases} f(\widetilde{x}_i, y) & \text{if } i < i^* \\ \widetilde{b} & \text{if } i = i^* - 1 \\ w & \text{otherwise} \end{cases},$$

   where $\widetilde{y}_i \leftarrow \mathcal{Y}$ *and* $\widetilde{x}_i \leftarrow \mathcal{X}$ *are independent.*
5. *For* $i = 1$ *to* r*:*
   (a) *The dealer sends* a_i *to* A*, which responds with either* continue *or* abort.
   (b) *If* A *responds with* abort*, then the dealer sends* abort *to* B *and halts, and* B *outputs* b_{i-1}.
   (c) *The dealer sends* b_i *to* B. *which responds with either* continue *or* abort.
   (d) *If* B *responds with* abort*, then the dealer sends* abort *to* A *and halts, and* A *outputs* a_i.
6. *If no party aborted, then* A *outputs* a_r *and* B *outputs* b_r.

Below we give a sufficient and necessary condition for the security of the protocol. We first introduce some new definitions. The first definition is the notion of an associated matrix for a 2-ary functionality.

Definition 5.6 (Associated matrix of 2-ary function)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a 2-ary Boolean function. We define its* associated matrix $\mathbf{M}_f \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{Y}|}$ *as follows. The rows and columns of $\mathbf{M}_f$ are indexed with the elements of $\mathcal{X}$ and $\mathcal{Y}$, respectively, and each entry is defined as $\mathbf{M}_f(x,y) = \Pr[f(x,y) = 1]$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We denote by $\overline{\mathbf{M}}_f$ the complement matrix of f, where $\overline{\mathbf{M}}_f = \mathbf{J} - \mathbf{M}_f$. Finally, we let $\mathbf{M}_{\mathsf{one}} = (\mathbf{M}_f||\mathbf{1})$ denote the matrix $\mathbf{M}_f$ concatenated with the all-one column. Finally, we index the last column by y_{one}, that is $\mathbf{M}_{\mathsf{one}}(\cdot, y_{\mathsf{one}}) = \mathbf{1}$.*

We now define an equivalence relation over the columns of a matrix. We are interested in when two columns have a non-zero entry in the same row of the reduced row echelon form of the matrix. Since this relation is not transitive, we define the equivalence relation as its transitive closure.

Definition 5.7. *Let $\mathbf{M} \in \mathbb{R}^{m\times\ell}$ be a matrix and let $\mathbf{R} = \mathsf{rref}(\mathbf{M})$ be its reduced row echelon form. For two columns $y, y' \in [\ell]$, we write $y \sim_{\mathbf{M}} y'$ if there exists $x \in [m]$ such that $\mathbf{R}(x,y) \neq 0$ and $\mathbf{R}(x,y') \neq 0$. We define the equivalence relation $\equiv_{\mathbf{M}}$ to be the transitive closure of $\sim_{\mathbf{M}}$. That is, $y \equiv_{\mathbf{M}} y'$ if there exists a sequence $y_1, \ldots, y_k \in [\ell]$ such that*

$$y \sim_{\mathbf{M}} y_1 \sim_{\mathbf{M}} \cdots \sim_{\mathbf{M}} y_k \sim_{\mathbf{M}} y'.$$

When $\mathbf{M}$ is clear from context, we write $y \equiv y'$ to alleviate notations.

We are now ready to state the main theorem of this section.

Theorem 5.8. *Let $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a Boolean two-party functionality, and let $\boldsymbol{\beta} \in [0,1]^{|\mathcal{Y}|}$. Assume that there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}$, and $\mathbf{p}^T \cdot \mathbf{1} = 1$, and that the following hold.*

1. *For all $y, y' \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_f} y'$ it holds that $\beta_y = \beta_{y'}$.*
2. *If $\mathbf{1} \in \mathrm{Im}(\mathbf{M}_f)$, then for every $y \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_{\mathsf{one}}} y_{\mathsf{one}}$ it holds that $\beta_y = 1$.*

Then, for all sufficiently small constant $\alpha > 0$, the protocol $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security.

Conversely, if for a constant $\alpha > 0$ the protocol $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security, then there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}$, and $\mathbf{p}^T \cdot \mathbf{1} = 1$, and for all $y, y \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_f} y'$ it holds that $\beta_y = \beta_{y'}$.

As a corollary, if $\mathbf{1} \notin \mathrm{Im}(\mathbf{M}_f)$ then Theorem 5.8 characterizes the set of functionalities that can be computed with full security using a protocol from $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$.

Corollary 5.9. *Let $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a Boolean two-party functionality such that $\mathbf{1} \notin \mathrm{Im}(\mathbf{M}_f)$, and let $\boldsymbol{\beta} \in [0,1]^{|\mathcal{Y}|}$. Then, there exists a constant $\alpha > 0$ for which $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security if and only if there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}$, and $\mathbf{p}^T \cdot \mathbf{1} = 1$, and for all $y, y' \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_f} y'$ it holds that $\beta_y = \beta_{y'}$.*

The proof of Theorem 5.8 is given below. Towards proving Theorem 5.8, we first prove three lemmas. The first two lemmas reduce each direction to a system of the form $\mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}$, where $\mathbf{B}$ is a diagonal matrix. The third lemma characterizes when such a system has a solution $\mathbf{X}$.

The first lemma reduces the security of the protocol into a system of linear equations.

Lemma 5.10. *Let $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a two-party Boolean functionality, let $\boldsymbol{\beta} \in [0,1]^{|\mathcal{Y}|}$, and let $\mathcal{N} = \mathcal{M} = \{\mathbf{M}_{\mathsf{one}}\}$. Assume that $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, (\boldsymbol{\beta}||1))$ is solvable. Then, for all sufficiently small constant $\alpha > 0$, the protocol $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security.*

Proof. Let $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ and $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ be the solutions for $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, (\boldsymbol{\beta}||1))$. That is,

$$\mathbf{X} \cdot \mathbf{M}_{\mathsf{one}} = \mathbf{M}_{\mathsf{one}} \cdot \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad \mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T, \quad \text{and} \quad \mathbf{p}^T \cdot \mathbf{1} = 1,$$

where $\mathbf{B} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ be the diagonal matrix for which the values on the main diagonal are β_y for all $y \in \mathcal{Y}$. We show that $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security. First, observe that correctness holds since $i^* > r$ occurs with negligible probability. It remains to show that the protocol is secure. Note that a corrupted B can be simulated easily. Intuitively, this is because A gets its backup value a_i before B gets its backup value b_i, hence A learns the output first. We formally handle this case in Appendix A.3. We next consider the case of a corrupted A. Fix a real-world adversary $\mathcal{A}$ that corrupts A. We define the simulator as follows.

1. Query the adversary for the input x it sends to the dealer.
2. Sample $i^* \leftarrow \mathrm{Geom}(\alpha)$ according to the geometric distribution with parameter α.
3. For $i = 1$ to $i^* - 1$:
   (a) Compute $a_i = f(x, \widetilde{y}_i)$, where $\widetilde{y}_i \leftarrow \mathcal{Y}$, and send it to $\mathcal{A}$.
   (b) If $\mathcal{A}$ aborts A then do the following.
      i. Sample $x^* \leftarrow \mathbf{v}_x^{a_i}$, where $\mathbf{v}_x^{a_i}$ is a probability vector to be defined by the analysis below.
      ii. Send x^* to the trusted party, output whatever $\mathcal{A}$ outputs, and halt.
4. Send x to the trusted party and receive $w = f(x, y)$.
5. For $i = i^*$ to r:
   - Send $a_i = w$ to $\mathcal{A}$.
   - If $\mathcal{A}$ aborts A, then output whatever it outputs and halt.
6. Output whatever $\mathcal{A}$ outputs and halt.

Let i denote the round where the adversary instructs A to abort (set to $r + 1$ if no such round exists). Note that if $i > i^*$, then the output of the honest party in both worlds is equal to $f(x, y)$, and the view of $\mathcal{A}$ is identically distributed given the output. Therefore, we may condition on the event $i \leq i^*$. Now, given that $i \leq i^*$, observe that the backup values $a_1, \ldots, a_{i-1}$ and b_{i-1} are independent in both worlds. Thus, it is enough to analyze the distribution of (a_i, b_{i-1}) in both worlds. In the

following, we fix the input $x \in \mathcal{X}$ that $\mathcal{A}$ sends to the dealer, let $s_x = \Pr_{\widetilde{y} \leftarrow \mathcal{Y}}[f(x, \widetilde{y}) = 1]$, and let $\mathbf{u} \in \mathbb{R}^{|\mathcal{X}|}$ denote the uniform probability vector over $\mathcal{X}$.

Let us first analyze the probability that $(a_i, b_{i-1}) = (1, 1)$. In the real world, it holds that

$$\begin{aligned}\Pr[(a_i, b_{i-1}) = (1,1) \mid i \leq i^*] &= (1-\alpha) \cdot \Pr[f(x, \widetilde{y}_i) = 1] \cdot \Pr[f(\widetilde{x}_i, y) = 1] \\ &\quad + \alpha \cdot \mathbf{M}_f(x, y) \cdot \beta_y \\ &= (1-\alpha) \cdot s_x \cdot \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \mathbf{M}_f(x, y) \cdot \beta_y.\end{aligned}$$

On the other hand, in the ideal world, it holds that

$$\begin{aligned}\Pr[(a_i, b_{i-1}) = (1,1) \mid i \leq i^*] &= (1-\alpha) \cdot \Pr[f(x, \widetilde{y}_i) = 1] \cdot \Pr[f(x^*, y) = 1 \mid a_i = 1] + \\ &\quad \alpha \cdot \mathbf{M}_f(x, y) \\ &= (1-\alpha) \cdot s_x \cdot \mathbf{v}_x^1 \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \mathbf{M}_f(x, y).\end{aligned}$$

For the protocol to be secure, it must hold that the distribution of (a_i, b_{i-1}) in the real world is identical to the distribution of (a_i, b_{i-1}) in the ideal world. Thus, it must hold that

$$\begin{aligned}&(1-\alpha) \cdot s_x \cdot \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \mathbf{M}_f(x, y) \cdot \beta_y \\ &= (1-\alpha) \cdot s_x \cdot \mathbf{v}_x^1 \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \mathbf{M}_f(x, y).\end{aligned} \tag{12}$$

Observe that if $s_x = 0$ then $f(x, \cdot) \equiv 0$ is the constant 0 function. Therefore, both sides of Equation (12) are 0. Assume now that $s_x \neq 0$. Isolating $\mathbf{v}_x^1$ results in

$$\mathbf{v}_x^1 \cdot \mathbf{M}_f(\cdot, y) = \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{M}_f(x, y) \cdot \beta_y - \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{M}_f(x, y).$$

Since this must hold for all $y \in \mathcal{Y}$, we get that

$$\mathbf{v}_x^1 \cdot \mathbf{M}_f = \mathbf{u}^T \cdot \mathbf{M}_f + \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot (\mathbf{M}_f(x, y) \cdot \beta_y)_{y \in \mathcal{Y}}^T - \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{e}_x^T \cdot \mathbf{M}_f, \tag{13}$$

where $\mathbf{e}_x$ is the x^{th} standard basis vector. Now, recall that by assumption, there exists a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ such that

$$\mathbf{X} \cdot \mathbf{M}_{\mathsf{one}} = \mathbf{M}_{\mathsf{one}} \cdot \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

Stated differently,

$$\mathbf{X} \cdot \mathbf{M}_f = \mathbf{M}_f \cdot \mathbf{B} \quad \text{and} \quad \mathbf{X} \cdot \mathbf{1} = \mathbf{1}.$$

Therefore Equation (13) may be written as

$$\mathbf{v}_x^1 \cdot \mathbf{M}_f = \left(\mathbf{u}^T + \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{X}(x, \cdot) - \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{e}_x^T \right) \cdot \mathbf{M}_f.$$

Setting $\mathbf{v}_x^1 = \mathbf{u}^T + \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{X}(x, \cdot) - \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{e}_x^T$ solves the equation. It remains to show that it is also a probability vector. First, observe that since $\mathbf{u}^T$ and $\mathbf{e}_x^T$ are probability vectors and since $\mathbf{X}(x, \cdot) \cdot \mathbf{1} = 1$, it holds that

$$\begin{aligned}\mathbf{v}_x^1 \cdot \mathbf{1} &= \mathbf{u}^T \cdot \mathbf{1} + \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot (\mathbf{X}(x, \cdot) \cdot \mathbf{1}) - \frac{\alpha}{(1-\alpha) \cdot s_x} \cdot \mathbf{e}_x^T \cdot \mathbf{1} \\ &= 1 + \frac{\alpha}{(1-\alpha) \cdot s_x} - \frac{\alpha}{(1-\alpha) \cdot s_x} \\ &= 1.\end{aligned}$$

Next, we show that $v_x^1(x') \geq 0$ for all $x' \in \mathcal{X}$. Indeed,

$$\begin{aligned} v_x^1(x') &\geq \frac{1}{|\mathcal{X}|} + \frac{\alpha}{(1-\alpha)\cdot s_x} \cdot \mathbf{X}(x,x') - \frac{\alpha}{(1-\alpha)\cdot s_x} \\ &= \frac{1}{|\mathcal{X}|} + \frac{\alpha}{(1-\alpha)\cdot s_x} \cdot (\mathbf{X}(x,x') - 1). \end{aligned}$$

Note that the second term tends to 0 as α tends to 0. Thus, $v_x^1(x')$ is arbitrarily close to $\frac{1}{|\mathcal{X}|} \geq 0$, hence it is positive for all sufficiently small constant $\alpha > 0$. We conclude that $\mathbf{v}_x^1$ is a probability vector.

It remains to consider the remaining three cases, i.e., the probability that $(a_i, b_{i-1}) = (0,0)$, that $(a_i, b_{i-1}) = (1,0)$, and that $(a_i, b_{i-1}) = (0,1)$. First, we argue that the case of $(a_i, b_{i-1}) = (0,0)$ can be analyzed using the same analysis of the case of $(a_i, b_{i-1}) = (1,1)$. To show this, we prove that there exists a matrix $\mathbf{X'} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ for which it holds that

$$\mathbf{X'} \cdot \overline{\mathbf{M}}_{\mathsf{one}} = \overline{\mathbf{M}}_{\mathsf{one}} \cdot \begin{pmatrix} \overline{\mathbf{B}} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}, \tag{14}$$

where $\overline{\mathbf{B}} = \mathbf{I} - \mathbf{B}$. The existence of $\mathbf{X'}$ will imply the case of $(a_i, b_{i-1}) = (0,0)$ by similar arguments as before.

Let $\mathbf{X'} = \mathbf{I} + \mathbf{P} - \mathbf{X}$, where $\mathbf{P} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ is the matrix that each row of it is equal to $\mathbf{p}^T$. Then,

$$\mathbf{X'} \cdot \mathbf{1} = \mathbf{I} \cdot \mathbf{1} + \mathbf{P} \cdot \mathbf{1} - \mathbf{X} \cdot \mathbf{1} = \mathbf{1} + \mathbf{1} - \mathbf{1} = \mathbf{1},$$

and for all $x \in \mathcal{X}$ it holds that

$$\begin{aligned} \mathbf{X'}(x,\cdot) \cdot \overline{\mathbf{M}}_f &= \mathbf{X'}(x,\cdot) \cdot (\mathbf{J} - \mathbf{M}_f) \\ &= \left(\mathbf{e}_x^T + \mathbf{p}^T - \mathbf{X}(x,\cdot)\right) \cdot (\mathbf{J} - \mathbf{M}_f) \\ &= \mathbf{e}_x^T \cdot \mathbf{J} - \mathbf{e}_x^T \cdot \mathbf{M}_f + \mathbf{p}^T \cdot \mathbf{J} - \mathbf{p}^T \cdot \mathbf{M}_f - \mathbf{X}(x,\cdot) \cdot \mathbf{J} + \mathbf{X}(x,\cdot) \cdot \mathbf{M}_f \\ &= \mathbf{1}^T - (\mathbf{M}_f(x,y))_{y\in\mathcal{Y}}^T + \mathbf{1}^T - \boldsymbol{\beta}^T - \mathbf{1}^T + (\mathbf{M}_f(x,y) \cdot \beta_y)_{y\in\mathcal{Y}}^T \\ &= (1 - \mathbf{M}_f(x,y) - \beta_y + \mathbf{M}_f(x,y) \cdot \beta_y)_{y\in\mathcal{Y}}^T \\ &= ((1 - \mathbf{M}_f(x,y)) \cdot (1 - \boldsymbol{\beta}))_{y\in\mathcal{Y}}^T \\ &= \left(\overline{\mathbf{M}}_f(x,y) \cdot (1 - \beta_y)\right)_{y\in\mathcal{Y}}^T. \end{aligned}$$

Finally, similarly to what we showed in the solitary output setting in Section 4.1, the case of $(a_i, b_{i-1}) = (0,1)$ is equivalent to the case of $(a_i, b_{i-1}) = (0,0)$ and the case of $(a_i, b_{i-1}) = (1,0)$ is equivalent to the case of $(a_i, b_{i-1}) = (1,1)$. To see this, first observe that

$$\Pr[a_i = 0] = \Pr[a_i = 0, b_{i-1} = 0] + \Pr[a_i = 0, b_{i-1} = 1].$$

Since a_i is identically distributed in both worlds and we proved that the probability that $(a_i, b_{i-1}) = (0,0)$ is the same in both worlds as well, it follows that the probability that $(a_i, b_{i-1}) = (0,1)$ is the same. Since all probabilities sum to 1, it follows that the probability that $(a_i, b_{i-1}) = (1,0)$ is also the same as the probability that $(a_i, b_{i-1}) = (1,1)$.

□

We next state and prove the second lemma, which states that the security of the protocol implies the existence of solutions $\mathbf{X}$ and $\mathbf{p}$ for a system of the form

$$\begin{cases} \mathbf{X} \cdot \mathbf{M}_f = \mathbf{M}_f \cdot \mathbf{B} \\ \mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T \\ \mathbf{p}^T \cdot \mathbf{1} = 1 \end{cases},$$

for a diagonal matrix $\mathbf{B} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$ and a vector $\boldsymbol{\beta} = \mathbf{B}(y, y)$ for all $y \in \mathcal{Y}$.

Lemma 5.11. *Let $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a Boolean two-party functionality, let $\alpha > 0$ be a constant, and let $\boldsymbol{\beta} = (\beta_y)_{y \in \mathcal{Y}} \in [0,1]^{|\mathcal{Y}|}$. Assume that $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security. Then $\mathbf{Sys}(\{\mathbf{M}_f\}, \{\mathbf{M}_f\}, \boldsymbol{\beta})$ is solvable.*

Proof. Fix $x \in \mathcal{X}$. We describe two real-world adversaries that corrupts A for $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$, and show that the existence of their simulators implies the statement of the lemma. The first adversary instructs A to abort before sending the first message if $a_1 = 1$, and otherwise instructs it to act honestly until the end of the protocol. The second adversary instructs A to abort after it obtains a_1. We first describe the first adversary and use it to show the existence of the solution matrix $\mathbf{X}$, and later we describe the second adversary and use it to show the existence of the solution vector $\mathbf{p}$. Let $\mathcal{A}$ be the adversary that corrupts A and works as follows.

1. Instruct A to send x to the dealer, and obtain the backup value a_1 from the dealer.
2. If $a_1 = 1$, then instruct A to abort the execution. The dealer then sends b_0 to B who outputs it.
3. Otherwise, if $a_1 \neq 1$ then instruct A to act honestly until the end of the protocol. The dealer then sends b_r to B who outputs it.

Denote by b the output of B. By the security assumption, there exists a simulator Sim (that may depend on κ) that simulates $\mathcal{A}$ in the ideal world. Let D_κ denote the distribution over $\mathcal{X}$ that is used to sample the input x^* sent by Sim to the trusted party. Denote by S the algorithm that on input $(\kappa, x, x^*, f(x^*, y))$, runs Sim given that it sent x^* to the trusted party, and outputs whatever it outputs.[10] Then assuming $x^* \leftarrow D_\kappa$, it follows that the output distribution of $\mathsf{S}(\kappa, x, x^*, f(x^*, y))$ is identical to the output distribution of the simulator (and the output of B is equal to $f(x^*, y)$). By the security assumption, it follows that

$$\{(\mathsf{S}(\kappa, x, x^*, f(x^*, y)), f(x^*, y))\}_{\kappa \in \mathbb{N}, y \in \mathcal{Y}} \stackrel{\mathrm{C}}{\equiv} \left\{\mathrm{REAL}_{\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta}), \mathcal{A}}(\kappa, x, y)\right\}_{\kappa \in \mathbb{N}, y \in \mathcal{Y}}.$$

Since the domain of f is of constant size, it follows that the ensembles are statistically close. In particular, it holds that

$$|\Pr[\mathsf{S}(\kappa, x, x^*, f(x^*, y)) = 1, f(x^*, y) = 1] - \Pr[a_1 = 1, b = 1]| = \mathsf{neg}(\kappa). \tag{15}$$

We now analyze the two probabilities, which shows that there exists a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ such that

$$\mathbf{X}(x, \cdot) \cdot \mathbf{M}_f = \mathbf{M}_f(x, \cdot) \cdot \mathbf{B} = (\mathbf{M}_f(x, y) \cdot \beta_y)_{y \in \mathcal{Y}}^T.$$

[10] Similarly to the solitary output setting, S might be inefficient.

We start by analyzing the real world. Let $\mathbf{u} \in \mathbb{R}^{|\mathcal{X}|}$ be the uniform probability vector over $\mathcal{X}$, and let $s_x = \Pr_{\widetilde{y} \leftarrow \mathcal{Y}}[f(x, \widetilde{y}) = 1]$. For every $y \in \mathcal{Y}$ it holds that

$$\begin{aligned}\Pr[a_1 = 1, b = 1] &= \Pr[1 < i^*] \cdot \Pr_{\widetilde{y} \leftarrow \mathcal{Y}}[f(x, \widetilde{y}) = 1] \cdot \Pr_{\widetilde{x} \leftarrow \mathcal{X}}[f(\widetilde{x}, y) = 1] \\ &\quad + \Pr[i^* = 1] \cdot \Pr[f(x, y) = 1] \cdot \beta_y \\ &= (1 - \alpha) \cdot s_x \cdot \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \mathbf{M}_f(x, y) \cdot \beta_y. \end{aligned} \tag{16}$$

We next analyze the probability that $a_1 = 1$ and $b = 1$ in the ideal world. Let $\mathbf{d}_\kappa \in [0,1]^{|\mathcal{X}|}$ denote the probability vector that corresponds to the distribution D_κ. For a possible output $a \in \{0,1\}$, let

$$q_{\kappa, x^*, a} = \Pr_{x^* \leftarrow D_\kappa}[\mathsf{S}(\kappa, x, x^*, a) = 1],$$

and let $\mathbf{Q}_{\kappa,a} \in [0,1]^{|\mathcal{X}| \times |\mathcal{X}|}$ be the diagonal matrix whose diagonals are defined as $\mathbf{Q}_{\kappa,a}(x^*, x^*) = q_{\kappa, x^*, a}$ for all $x^* \in \mathcal{X}$. Observe that

$$\begin{aligned}\Pr[(\mathsf{S}(\kappa, x, x^*, f(x^*, y)) = 1, b = 1)] &= \sum_{x^* \in \mathcal{X}} d_\kappa(x^*) \cdot \mathbf{M}_f(x^*, y) \cdot q_{\kappa, x^*, 1} \\ &= \mathbf{d}_\kappa^T \cdot \mathbf{Q}_{\kappa,1} \cdot \mathbf{M}_f \cdot \mathbf{e}_y,\end{aligned}$$

where $\mathbf{e}_y$ is the y^{th} standard basis vector.

Combined with Equations (15) and (16), we get that

$$\left|(1 - \alpha) \cdot s_x \cdot \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \mathbf{M}_f(x, y) \cdot \beta_y - \mathbf{d}_\kappa^T \cdot \mathbf{Q}_{\kappa,1} \cdot \mathbf{M}_f \cdot \mathbf{e}_y\right| \leq \mathsf{neg}(\kappa),$$

for all $y \in \mathcal{Y}$. Therefore, the vector

$$\frac{1}{\alpha} \cdot \left((1 - \alpha) \cdot s_x \cdot \mathbf{u}^T - \mathbf{d}_\kappa^T \cdot \mathbf{Q}_{\kappa,1}\right) \cdot \mathbf{M}_f$$

tends to $(\mathbf{M}_f(x, y) \cdot \beta_y)_{y \in \mathcal{Y}}$ as $\kappa \to \infty$. Recall that α is constant and all entries in the two vectors are probabilities, hence the preimage belongs to $[-1/\alpha, 1/\alpha]^{|\mathcal{X}|}$. Since this set is mapped by $\mathbf{M}_f$ to a closed set (in the topological sense), there exists $\mathbf{x}$ such that $\mathbf{x} \cdot \mathbf{M}_f = (\mathbf{M}_f(x, y) \cdot \beta_y)_{y \in \mathcal{Y}}$. Since it holds for all $x \in \mathcal{X}$, it holds that there exists $\mathbf{X}$ such that

$$\mathbf{X} \cdot \mathbf{M}_f = \mathbf{M}_f \cdot \mathbf{B}.$$

To conclude the proof we show that there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T$ and $\mathbf{p}^T \cdot \mathbf{1} = 1$. Fix a real-world adversary $\mathcal{A}_1$ that corrupts A, sends x to the dealer, and instructs A to abort after receiving the backup value a_1 from the dealer. Let $w(y)$ denote the output of B in the real-world when holds input y. Then for all inputs $y \in \mathcal{Y}$ it holds that

$$\begin{aligned}\Pr[w(y) = 1] &= \Pr[1 < i^*] \cdot \Pr[f(\widetilde{x}_i, y) = 1] + \Pr[1 = i^*] \cdot \beta_y \\ &= (1 - \alpha) \cdot \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \beta_y.\end{aligned}$$

In the ideal world, a simulator sends to the trusted party a value that depends on x. Therefore, if $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security, there must exist a probability vector $\mathbf{x}_x \in \mathbb{R}^{|\mathcal{X}|}$ such that

$$\mathbf{x}_x^T \cdot \mathbf{M}_f(\cdot, y) = (1 - \alpha) \cdot \mathbf{u}^T \cdot \mathbf{M}_f(\cdot, y) + \alpha \cdot \beta_y.$$

Since it must hold for all $y \in \mathcal{Y}$, it must hold that

$$\mathbf{x}_x^T \cdot \mathbf{M}_f = (1-\alpha) \cdot \mathbf{u}^T \cdot \mathbf{M}_f + \alpha \cdot \boldsymbol{\beta}^T.$$

Rewriting the equation results in

$$\frac{1}{\alpha}\left(\mathbf{x}_x^T - (1-\alpha) \cdot \mathbf{u}^T\right) \cdot \mathbf{M}_f = \boldsymbol{\beta}^T.$$

Observe that for $\mathbf{p}^T = \frac{1}{\alpha}\left(\mathbf{x}_x^T - (1-\alpha) \cdot \mathbf{u}^T\right)$ it holds that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T$, and that

$$\begin{aligned}
\mathbf{p}^T \cdot \mathbf{1} &= \frac{1}{\alpha}\left(\mathbf{x}_x^T - (1-\alpha) \cdot \mathbf{u}^T\right) \cdot \mathbf{1} \\
&= \frac{1}{\alpha}\left(\mathbf{x}_x^T \cdot \mathbf{1} - (1-\alpha) \cdot \mathbf{u}^T \cdot \mathbf{1}\right) \\
&= \frac{1}{\alpha}\left(1 - (1-\alpha)\right) \\
&= 1.
\end{aligned}$$

Therefore, $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, \boldsymbol{\beta})$ is solvable.

□

We now state and prove the third lemma characterizing when a system of the form $\mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}$, where $\mathbf{B}$ is a diagonal matrix, has a solution $\mathbf{X}$.

Lemma 5.12. *Let $\mathbf{M} \in \mathbb{R}^{m \times \ell}$ and let $\boldsymbol{\beta} \in [0,1]^\ell$. Define $\mathbf{B} \in \mathbb{R}^{\ell \times \ell}$ to be the diagonal matrix whose y^{th} entry on the diagonal is β_y. Then there exist $\mathbf{X} \in \mathbb{R}^{m \times m}$ such that $\mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}$ if and only if for all $y, y' \in [\ell]$ such that $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$.*

Proof. We first transform the system of equations into an equivalent system. Let $\widehat{\mathbf{R}} = \mathsf{rref}(\mathbf{M})$, and let $\mathbf{E} \in \mathbb{R}^{m \times m}$ be the matrix that transforms $\mathbf{M}$ into $\widehat{\mathbf{R}}$ using elementary row operations. That is, $\mathbf{E} \cdot \mathbf{M} = \widehat{\mathbf{R}}$. Since row reduction is an invertible process, there exists matrix $\mathbf{E}^{-1} \in \mathbb{R}^{m \times m}$ such that $\mathbf{E}^{-1} \cdot \mathbf{E} = \mathbf{I}$. Let $\widehat{\mathbf{X}} = \mathbf{X} \cdot \mathbf{E}^{-1}$. Therefore,

$$\widehat{\mathbf{X}} \cdot \widehat{\mathbf{R}} = \mathbf{X} \cdot \mathbf{E}^{-1} \cdot \mathbf{E} \cdot \mathbf{M} = \mathbf{X} \cdot \mathbf{M} = \mathbf{M} \cdot \mathbf{B}.$$

Multiplying both sides of the equation by $\mathbf{E}$ and letting $\widetilde{\mathbf{X}} = \mathbf{E} \cdot \widehat{\mathbf{X}}$ results in

$$\widetilde{\mathbf{X}} \cdot \widehat{\mathbf{R}} = \widehat{\mathbf{R}} \cdot \mathbf{B}.$$

Let $\mathbf{R} = \mathsf{rref}^*(\mathbf{M})$. Observe that a solution $\widetilde{\mathbf{X}}$ exists if and only if there exists $\mathbf{X'} \in \mathbb{R}^{m \times m'}$ such that

$$\mathbf{X'} \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{B}. \tag{17}$$

Indeed, since $\mathbf{R}$ is the matrix $\widehat{\mathbf{R}}$ with the zero-rows removed, we get that a solution $\widetilde{\mathbf{X}}$ exists if and only if $\mathbf{X'}$ exists. For the proof, we apply Lemma 5.2, which characterizes for what vectors

$\boldsymbol{\beta}$ there exists a solution for the equation. Let $m' = \mathsf{rank}(\mathbf{R})$ and define $\mathbf{K} \in \mathbb{R}^{(m\cdot(\ell-m'))\times\ell}$ as in Lemma 5.2, that is

$$\mathbf{K}((x, y_{\mathsf{free}}), y) = \begin{cases} \mathbf{R}(x,y)\cdot\mathbf{R}(\mathsf{ptr}_{\mathbf{R}}(y), y_{\mathsf{free}}) & \text{if } y \in \mathcal{P}_{\mathbf{R}} \\ -\mathbf{R}(x,y) & \text{if } y = y_{\mathsf{free}} \\ 0 & \text{otherwise} \end{cases},$$

for all $x \in [m']$, $y_{\mathsf{free}} \in \mathcal{F}_{\mathbf{R}}$, and $y \in [\ell]$. Then Equation (17) has a solution $\mathbf{X}'$ if and only if $\boldsymbol{\beta} \in \mathrm{Ker}(\mathbf{K})$. To conclude the proof, we characterize $\mathrm{Ker}(\mathbf{K})$.

Fix $x \in [m']$, $y_{\mathsf{free}} \in \mathcal{F}_{\mathbf{R}}$ and $y \in [\ell]$, and consider the following cases.

- If $y \in \mathcal{P}_{\mathbf{R}}$, then there exists $x' \in [m']$ such that $\mathbf{R}(x', y) = 1$, and for all $\widetilde{x} \neq x'$, it holds that $\mathbf{R}(\widetilde{x}, y) = 0$. It follows that if $x = \mathsf{ptr}_{\mathbf{R}}(y)$ then $\mathbf{K}((x, y_{\mathsf{free}}), y) = 1 \cdot \mathbf{R}(x, y_{\mathsf{free}}) = \mathbf{R}(x, y_{\mathsf{free}})$, and if $x \neq \mathsf{ptr}_{\mathbf{R}}(y)$ then $\mathbf{K}((x, y_{\mathsf{free}}), y) = 0 \cdot \mathbf{R}(x, y_{\mathsf{free}}) = 0$.
- If $y = y_{\mathsf{free}}$ then $\mathbf{K}((x, y_{\mathsf{free}}), y) = -\mathbf{R}(x, y_{\mathsf{free}})$.
- In all other cases, $\mathbf{K}((x, y_{\mathsf{free}}), y) = 0$, by the definition of $\mathbf{K}$.

Then, if $\mathbf{R}(x, y_{\mathsf{free}}) \neq 0$, the only non-zero entries of the row (x, y_{free}) are in the columns $y = \mathsf{ptr}_{\mathbf{R}}^{-1}(x)$ and $y = y_{\mathsf{free}}$. Furthermore, since $\mathbf{R}(x, y_{\mathsf{free}}) \neq 0$ and $\mathbf{R}(x, \mathsf{ptr}_{\mathbf{R}}^{-1}(x)) = 1$, it follows that $y_{\mathsf{free}} \sim \mathsf{ptr}_{\mathbf{R}}^{-1}(x)$. We conclude that $\boldsymbol{\beta} \in \mathrm{Ker}(\mathbf{K})$ if and only if $\beta_{\mathsf{ptr}_{\mathbf{R}}^{-1}(x)} = \beta_{y_{\mathsf{free}}}$ for all x and y_{free} such that $\mathsf{ptr}_{\mathbf{R}}^{-1}(x) \sim y_{\mathsf{free}}$. To complete the proof, we show that for all $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$. First, observe that for every column y such that $\mathbf{R}(\cdot, y) = \mathbf{0}$, it holds that y is equivalent to itself, and hence the statement trivially holds. Now, we stress that for every free column y_{free}, it holds that there exists a pivot column y_{piv} such that $y_{\mathsf{free}} \sim y_{\mathsf{piv}}$. This is because if no such y_{piv} it holds that y_{free} is a pivot column itself. The above analysis shows that for such y_{free} and y_{piv} it holds that $\beta_{y_{\mathsf{free}}} = \beta_{y_{\mathsf{piv}}}$. Thus, by inductive argument, for all columns y and y' such that $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$.

□

We are now ready to prove Theorem 5.8 using Lemmas 5.10 to 5.12.

Proof of Theorem 5.8. First, assume that there exists a constant $\alpha > 0$ for which $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security. By Lemma 5.11, there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T$ and $\mathbf{p}^T \cdot \mathbf{1} = 1$, and a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ such that $\mathbf{X} \cdot \mathbf{M}_f = \mathbf{M}_f \cdot \mathbf{B}$. By Lemma 5.12, such $\mathbf{X}$ exists if and only if for all $y, y' \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_f} y'$ it holds that $\beta_y = \beta_{y'}$.

Now, assume that there exists a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ such that $\mathbf{p}^T \cdot \mathbf{M}_f = \boldsymbol{\beta}^T$ and $\mathbf{p}^T \cdot \mathbf{1} = 1$, and that for all $y, y' \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_f} y'$ it holds that $\beta_y = \beta_{y'}$, and assume that if $\mathbf{1} \in \mathrm{Im}(\mathbf{M}_f)$, then for all $y \in \mathcal{Y}$ such that $y \equiv_{\mathbf{M}_{\mathsf{one}}} y_{\mathsf{one}}$, it holds that $\beta_y = 1$. First, observe that for two distinct values $y, y' \in \mathcal{Y}$, it holds that $y \equiv_{\mathbf{M}_f} y'$ if and only if $y \equiv_{\mathbf{M}_{\mathsf{one}}} y'$. This is because the pivot columns in the reduced row echelon form of $\mathbf{M}_f$, will remain pivot columns in the reduced row echelon form of $\mathbf{M}_{\mathsf{one}}$. Note that the rank of $\mathbf{M}_{\mathsf{one}}$ is not necessarily equal to the rank of $\mathbf{M}_f$, but the pivot columns that are associated with the columns of $\mathbf{M}_f$ in $\mathbf{M}_{\mathsf{one}}$ remain the same as in the reduced row echelon form of $\mathbf{M}_f$. Thus, for the rest of this proof, we write $y \equiv y'$ instead of $y \equiv_{\mathbf{M}_f} y'$ and $y \equiv_{\mathbf{M}_{\mathsf{one}}} y'$.

To show that $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security (for all sufficiently small $\alpha > 0$) by Lemma 5.10, it suffices to show the existence of a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ such that

$$\mathbf{X} \cdot \mathbf{M}_{\mathsf{one}} = \mathbf{M}_{\mathsf{one}} \cdot \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

Let $\mathbf{R} = \mathsf{rref}^*(\mathbf{M}_{\mathsf{one}})$. We separate the proof into two cases. For the first case, we assume that $\mathbf{1} \notin \mathrm{Im}(\mathbf{M}_f)$. Then, $\mathsf{rank}(\mathbf{M}_f) < \mathsf{rank}(\mathbf{M}_{\mathsf{one}})$, which implies that $y_{\mathsf{one}} \in \mathcal{P}_{\mathbf{R}}$. Therefore, there is no $y \in \mathcal{Y}$ for which it holds that $y \equiv y_{\mathsf{one}}$. Thus, by Lemma 5.12, there exists a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ such that

$$\mathbf{X} \cdot \mathbf{M}_{\mathsf{one}} = \mathbf{M}_{\mathsf{one}} \cdot \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

For the second case, we assume that $\mathbf{1} \in \mathrm{Im}(\mathbf{M}_f)$. Then $\mathsf{rank}(\mathbf{M}_f) = \mathsf{rank}(\mathbf{M}_{\mathsf{one}})$, which implies that $y_{\mathsf{one}} \in \mathcal{F}_{\mathbf{R}}$. Recall that we assume that for all $y \in \mathcal{Y}$ such that $y \equiv y_{\mathsf{one}}$ it holds that $\beta_y = 1$, and for all $y, y' \in \mathcal{Y}$ such that $y \equiv y'$ it holds that $\beta_y = \beta_{y'}$. Therefore, by Lemma 5.12 there exists a matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ such that

$$\mathbf{X} \cdot \mathbf{M}_{\mathsf{one}} = \mathbf{M}_{\mathsf{one}} \cdot \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

In both cases, since $\mathbf{Sys}(\mathcal{N}, \mathcal{M}, (\boldsymbol{\beta}||1))$ (for $\mathcal{N} = \mathcal{M} = \{\mathbf{M}_{\mathsf{one}}\}$) is solvable, by Lemma 5.10 for all sufficiently small constant $\alpha > 0$, the protocol $\pi_{\mathsf{sr}}^{\mathsf{fair}}(\alpha, \boldsymbol{\beta})$ computes f with full security. □

6 Generalizing the AOV Impossibility Result

In this chapter, we extend the impossibility result of [7] to the setting where C has an input. Similarly to [7], we consider only Boolean functionalities. In order to state and understand the result, we first need to generalize the notion of *locking strategies* [31, 21, 32] to the setting where a third party holds an input. Originally, locking strategies were defined to capture two-party functionalities that can be used to construct fair sampling protocols, which allow two parties to sample dependent values – a task that is known to be impossible to do [17, 2].

Roughly speaking, a locking strategy (for two-party functionalities) for a party P is a way for P to sample an input, and apply a local operation to the output of the function, such that the distribution of its final output is independent of the other party's input.

For Boolean functions, the local operation can be either to flip the output or to keep it, possibly depending on the input. Makriyannis [31] showed that these strategies (for each party) can be encoded using a single real-valued vector. The absolute value of each entry represents the weight of the corresponding input, and the sign represents whether or not the party should flip the output (given the input). Alon et al. [7] used locking strategies in order to identify the set of all Boolean 2-ary solitary output three-party functionalities that cannot be computed with full security (dubbing them strong semi-balanced).

We generalize the notion of locking strategies to the setting where the input of the output-receiving party C is fixed (and known) to some value z. We refer to these strategies as *z-locking strategies*. In essence, these are locking strategies to the induced two-party functionality, when C holds input z. Formally, we define them for Boolean functionalities as follows.

Definition 6.1 (z-Locking strategies for Boolean functionalities)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a Boolean three-party functionality and fix $z \in \mathcal{Z}$. A* z-locking strategy *for A is a vector $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ satisfying $\mathbf{p}^T \cdot \mathbf{M}_z = \delta_1 \cdot \mathbf{1}^T$, for some value $\delta_1 \in \mathbb{R}$. Similarly, a z-locking strategy for B is a vector $\mathbf{q} \in \mathbb{R}^{|\mathcal{Y}|}$ satisfying $\mathbf{M}_z \cdot \mathbf{q} = \delta_2 \cdot \mathbf{1}$, for some value $\delta_2 \in \mathbb{R}$.*

Next, we generalize the notion of *strong semi-balanced* functionalities introduced by [7] to the setting where C has an input. We later show that strong semi-balanced functionalities cannot be securely computed.

Definition 6.2 (Strong semi-balanced functionalities)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a solitary output Boolean three-party functionality. We call f* strong semi-balanced*, if there exist vectors $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ and $\mathbf{q} \in \mathbb{R}^{|\mathcal{Y}|}$ for A and B, respectively, and there exists $z \in \mathcal{Z}$, such that for every $z' \in \mathcal{Z}$, for every $x \in \mathcal{X}$, and for every $y \in \mathcal{Y}$, it holds that*

$$\begin{cases} \mathbf{p}^T \cdot \mathbf{M}_z = \mathbf{1}^T, \\ \mathbf{1}^T \cdot \mathbf{p} < 1, \\ -1 + \mathbf{1}^T \cdot \mathbf{p} \leq \mathbf{p}^T \cdot \mathbf{M}_{z'}(\cdot, y) \leq 1 \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{M}_z \cdot \mathbf{q} = \mathbf{1}^T, \\ \mathbf{1}^T \cdot \mathbf{q} < 1, \\ -1 + \mathbf{1}^T \cdot \mathbf{q} \leq \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} \leq 1 \end{cases}.$$

Intuitively, $\mathbf{p}$ and $\mathbf{q}$ are z-locking strategies satisfying an additional relation with other possible inputs for C, which is the third part of the definition. We now show that if a solitary output Boolean three-party functionality f is strong semi-balanced, then f cannot be computed with full security.

Theorem 6.3. *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a strong semi-balanced solitary output Boolean three-party functionality. Then f cannot be computed with full security.*

Proof. Assume towards a contradiction that there exists a secure r-round protocol π computing f. Since f is strong semi-balanced, there exist vectors $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ and $\mathbf{q} \in \mathbb{R}^{|\mathcal{Y}|}$ such that

$$\begin{cases} \mathbf{p}^T \cdot \mathbf{M}_z = \delta_1 \cdot \mathbf{1}^T, \text{ where } \delta_1 > 0 \\ \mathbf{1}^T \cdot \mathbf{p} < \delta_1, \\ \sum_{x \in \mathcal{X}} |p_x| = 1, \\ -\delta_1 + \mathbf{1}^T \cdot \mathbf{p} \leq \mathbf{p}^T \cdot \mathbf{M}_{z'}(\cdot, y) \leq \delta_1 \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{M}_z \cdot \mathbf{q} = \delta_2 \cdot \mathbf{1}, \text{ where } \delta_2 > 0 \\ \mathbf{1}^T \cdot \mathbf{q} < \delta_2, \\ \sum_{y \in \mathcal{Y}} |q_y| = 1, \\ -\delta_2 + \mathbf{1}^T \cdot \mathbf{q} \leq \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} \leq \delta_2 \end{cases}.$$

To see this, fix z-locking strategies $\widetilde{\mathbf{p}}$ and $\widetilde{\mathbf{q}}$ guaranteed to exist by our assumption that f is strong semi-balanced. Then define $\mathbf{p}$ and $\mathbf{q}$ to be $\widetilde{\mathbf{p}}$ and $\widetilde{\mathbf{q}}$ normalized with respect to the ℓ_1 norm, respectively. That is, for $\delta_1 = (\sum_{x \in \mathcal{X}} |\widetilde{p}_x|)^{-1}$ and $\delta_2 = (\sum_{y \in \mathcal{Y}} |\widetilde{q}_y|)^{-1}$, define $\mathbf{p} = \delta_1 \cdot \widetilde{\mathbf{p}}$ and $\mathbf{q} = \delta_2 \cdot \widetilde{\mathbf{q}}$.

Consider an execution of π, where the input of C is the value z, which satisfies $\mathbf{p}^T \cdot \mathbf{M}_z = \delta_1 \cdot \mathbf{1}^T$ and $\mathbf{M}_z \cdot \mathbf{q} = \delta_2 \cdot \mathbf{1}$. We further assume that the inputs x and y of A and B, respectively, are sampled independently according to $|\mathbf{p}|$ and $|\mathbf{q}|$, respectively. That is, A holds the input x with probability $|p_x|$ and B holds the input y with probability $|q_y|$. We next introduce some notations. Let $\mathsf{flip}(x)$ output 1 if $p_x < 0$, and 0 otherwise. Similarly, let $\mathsf{flip}(y)$[11] output 1 if $q_y < 0$, and 0 otherwise. Let $p^- = \sum_{x \in \mathcal{X} : p_x < 0} |p_x|$ denote the probability that $\mathsf{flip}(x) = 1$ and let $q^- = \sum_{y \in \mathcal{Y} : q_y < 0} |q_y|$ denote the probability that $\mathsf{flip}(y) = 1$. We further let $p^+ = 1 - p^-$ and $q^+ = 1 - q^-$. Observe that $\mathbf{1}^T \cdot \mathbf{p} = p^+ - p^-$ and that $\mathbf{1}^T \cdot \mathbf{q} = q^+ - q^-$. We use the following lemma proved by [7], stating that there exists a real world adversary that can guess either $\mathsf{flip}(y)$ or $\mathsf{flip}(x)$ with "good" probability.

[11] Formally, we assume without loss of generality that the domains $\mathcal{X}$ and $\mathcal{Y}$ are disjoint.

Lemma 6.4 ([7, Lemma 4.2])**.** *There exists a constant $\xi > 0$ (independent of the protocol) such that one of the following holds.*

- *There exists a real-world adversary corrupting* A *and* C *that can guess* $\mathsf{flip}(y)$ *with probability at least* $\delta_2 + q^- + \xi/r$.
- *There exists a real-world adversary corrupting* B *and* C *that can guess* $\mathsf{flip}(x)$ *with probability at least* $\delta_1 + p^- + \xi/r$.

At first, it may seem like we cannot apply their lemma in our setting since they assumed that C does not have an input. However, since we fixed the input of C, the resulting execution of the protocol computes the 2-ary functionality $f_z(x, y, \lambda) = f(x, y, z)$. Now, similarly to [7], we can use the security of π against adversaries corrupting either A or B (with an honest C in both cases) to complete the proof. For completeness, we provide the description of the attack and the proof of Lemma 6.4 in Appendix A.1.

Without loss of generality, we may assume that the first item of the lemma holds. Let $\mathcal{A}$ be the real-world adversary corrupting A and C guaranteed to exist by Lemma 6.4 (i.e., $\mathcal{A}$ can guess $\mathsf{flip}(y)$ with probability at least $\delta_2 + q^- + \xi/r$). We next show that no ideal-world simulator can guess this value as well as the real-world adversary. That is, we prove the following claim.

Claim 6.5. *For any (possibly randomized) algorithm* $\mathsf{S} : \mathcal{X} \times \mathcal{Z} \times \{0,1\} \to \{0,1\}$, *every* $x \in \mathcal{X}$, *and every* $z' \in \mathcal{Z}$, *it holds that*

$$\Pr_{y \leftarrow |\mathbf{q}|}[\mathsf{S}(x, z', f(x, y, z')) = \mathsf{flip}(y)] \leq \delta_2 + q^-.$$

Proof. Fix $x \in \mathcal{X}$ and $z' \in \mathcal{Z}$, and let $w = f(x, y, z')$, and $\widetilde{w} = w \oplus \mathsf{flip}(y)$. For brevity, we write $\mathsf{S}(w)$ instead of $\mathsf{S}(x, z', w)$. It holds that

$$\begin{aligned}
\Pr[\mathsf{S}(w) = \mathsf{flip}(y)] &= \Pr[\mathsf{S}(w) \oplus w = \widetilde{w}] \\
&= \Pr[\mathsf{S}(0) = 0, w = 0, \widetilde{w} = 0] + \Pr[\mathsf{S}(1) = 1, w = 1, \widetilde{w} = 0] \\
&\quad + \Pr[\mathsf{S}(0) = 1, w = 0, \widetilde{w} = 1] + \Pr[\mathsf{S}(1) = 0, w = 1, \widetilde{w} = 1] \\
&= \Pr[\mathsf{S}(0) = 0] \cdot \Pr[w = 0, \widetilde{w} = 0] + \Pr[\mathsf{S}(1) = 1] \cdot \Pr[w = 1, \widetilde{w} = 0] \\
&\quad + \Pr[\mathsf{S}(0) = 1] \cdot \Pr[w = 0, \widetilde{w} = 1] + \Pr[\mathsf{S}(1) = 0] \cdot \Pr[w = 1, \widetilde{w} = 1] \\
&\leq \max\{\Pr[w = 0, \widetilde{w} = 0], \Pr[w = 0, \widetilde{w} = 1]\} \\
&\quad + \max\{\Pr[w = 1, \widetilde{w} = 0], \Pr[w = 1, \widetilde{w} = 1]\}.
\end{aligned}$$

Depending on which quantities are larger, the above expression is upper-bounded by one of the following.

- $\Pr[\widetilde{w} = w] = \Pr[\mathsf{flip}(y) = 0] = q^+$,
- $\Pr[\widetilde{w} \neq w] = \Pr[\mathsf{flip}(y) = 1] = q^-$,
- $\Pr[\widetilde{w} = 0]$,
- $\Pr[\widetilde{w} = 1]$,

where the last equality in the first two equations is implied by the definitions of q^+ and q^-. Observe that $q^- < \delta_2 + q^-$ since $\delta_2 > 0$. Additionally since we assume that $\mathbf{1}^T \cdot \mathbf{q} < \delta_2$, it follows that

$$\delta_2 + q^- > \mathbf{1}^T \cdot \mathbf{q} + q^- = q^+ - q^- + q^- = q^+.$$

Therefore $q^+ < \delta_2 + q^-$.

It remains to show that $\Pr[\widetilde{w} = 1]$ and $\Pr[\widetilde{w} = 0]$ are also upper bounded by $\delta_2 + q^-$. Indeed, it holds that

$$\begin{aligned}
\Pr[\widetilde{w} = 1] &= \Pr[w = 1, \mathsf{flip}(y) = 0] + \Pr[w = 0, \mathsf{flip}(y) = 1] \\
&= \sum_{y:\, q_y > 0} \mathbf{M}_{z'}(x, y) \cdot q_y + \sum_{y:\, q_y < 0} (1 - \mathbf{M}_{z'}(x, y)) \cdot (-q_y) \\
&= \sum_{y} \mathbf{M}_{z'}(x, y) \cdot q_y + q^- \\
&= \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} + q^- \\
&\leq \delta_2 + q^-,
\end{aligned}$$

where the inequality holds since we assume that $\mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} \leq \delta_2$. Finally, observe that since the above equation also proves that $\Pr[\widetilde{w} = 1] = \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} + q^-$, it follows that

$$\begin{aligned}
\Pr[\widetilde{w} = 0] &= 1 - \Pr[\widetilde{w} = 1] = 1 - (\mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} + q^-) \\
&\leq 1 - (-\delta_2 + \mathbf{1}^T \cdot \mathbf{q} + q^-) \\
&= 1 + \delta_2 - \mathbf{1}^T \cdot \mathbf{q} - q^- \\
&= \delta_2 + q^+ + q^- - q^+ \\
&= \delta_2 + q^-,
\end{aligned}$$

where the inequality holds since $-\delta_2 + \mathbf{1}^T \cdot \mathbf{q} \leq \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q}$, and the third equality is due to the fact that $q^- = 1 - q^+$ and that $\mathbf{1}^T \cdot \mathbf{q} = q^+ - q^-$. □

We get that every ideal world simulator can guess $\mathsf{flip}(y)$ with a probability at most $\delta_2 + q^-$. Thus, the adversary $\mathcal{A}$ cannot be simulated and we get a contradiction. □

Bibliography

[1] N. Agarwal, S. Anand, and M. Prabhakaran. Uncovering algebraic structures in the MPC landscape. In Y. Ishai and V. Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 381–406. Springer, 2019.

[2] S. Agrawal and M. Prabhakaran. On fair exchange, fair coins and fair sampling. In *CRYPTO*, pages 259–276, 2013.

[3] B. Alon and E. Omri. On secure computation of solitary output functionalities with and without broadcast. In G. N. Rothblum and H. Wee, editors, *Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part II*, volume 14370 of *Lecture Notes in Computer Science*, pages 94–123. Springer, 2023.

[4] B. Alon, A. Beimel, and E. Omri. Three party secure computation with friends and foes. In G. N. Rothblum and H. Wee, editors, *Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part II*, volume 14370 of *Lecture Notes in Computer Science*, pages 156–185. Springer, 2023.

[5] B. Alon, R. Cohen, E. Omri, and T. Suad. On the power of an honest majority in three-party computation without broadcast. *J. Cryptol.*, 36(3):25, 2023.

[6] B. Alon, M. Naor, E. Omri, and U. Stemmer. MPC for tech giants (GMPC): enabling gulliver and the lilliputians to cooperate amicably. In L. Reyzin and D. Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VIII*, volume 14927 of *Lecture Notes in Computer Science*, pages 74–108. Springer, 2024. doi: 10.1007/978-3-031-68397-8_3. URL https://doi.org/10.1007/978-3-031-68397-8_3.

[7] B. Alon, E. Omri, and M. Venkitasubramaniam. Can alice and bob guarantee output to carol? In M. Joye and G. Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part V*, volume 14655 of *Lecture Notes in Computer Science*, pages 32–61. Springer, 2024.

[8] G. Asharov. Towards characterizing complete fairness in secure two-party computation. In *TCC*, pages 291–316, 2014.

[9] G. Asharov, A. Beimel, N. Makriyannis, and E. Omri. Complete characterization of fairness in secure two-party computation of Boolean functions. In *Proceedings of the 12th Theory of Cryptography Conference(TCC), part I*, pages 199–228, 2015.

[10] A. Beimel, Y. Lindell, E. Omri, and I. Orlov. $1/p$-secure multiparty computation without honest majority and the best of both worlds. In *30th Annual International Cryptology Conference (CRYPTO)*, pages 277–296, 2011.

[11] A. Beimel, A. Gabizon, Y. Ishai, E. Kushilevitz, S. Meldgaard, and A. Paskin-Cherniavsky. Non-interactive secure multiparty computation. In J. A. Garay and R. Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 387–404. Springer, 2014.

[12] A. Beimel, E. Omri, and I. Orlov. Protocols for multiparty coin toss with a dishonest majority. *Journal of Cryptology*, 28(3):551–600, 2015.

[13] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova. Secure single-server aggregation with (poly)logarithmic overhead. In J. Ligatti, X. Ou, J. Katz, and G. Vigna, editors, *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*, pages 1253–1269. ACM, 2020.

[14] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 1988.

[15] K. A. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1175–1191. ACM, 2017.

[16] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.

[17] R. Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *Proceedings of the 18th Annual ACM STOC*, 1986.

[18] R. Cohen and Y. Lindell. Fairness versus guaranteed output delivery in secure multiparty computation. *Journal of Cryptology*, 30(4):1157–1186, 2017.

[19] R. Cohen, I. Haitner, E. Omri, and L. Rotem. Characterization of secure multiparty computation without broadcast. *JoC*, 31(2):587–609, 2018.

[20] D. Dachman-Soled. Revisiting fairness in MPC: polynomial number of parties and general adversarial structures. In R. Pass and K. Pietrzak, editors, *Proceedings of the 18th Theory of Cryptography Conference(TCC), part II*, volume 12551, pages 595–620. Springer, 2020.

[21] V. Daza and N. Makriyannis. Designing fully secure protocols for secure two-party computation of constant-domain functions. In *Proceedings of the 15th Theory of Cryptography Conference(TCC), part I*, pages 581–611, 2017.

[22] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation (extended abstract). In F. T. Leighton and M. T. Goodrich, editors, *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 554–563. ACM, 1994.

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

[24] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the 51st Annual ACM STOC*, pages 218–229, 1987.

[25] S. D. Gordon and J. Katz. Complete fairness in multi-party computation without an honest majority. In *Proceedings of the 6th Theory of Cryptography Conference(TCC)*, pages 19–35, 2009.

[26] S. D. Gordon, C. Hazay, J. Katz, and Y. Lindell. Complete fairness in secure two-party computation. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 413–422, 2008.

[27] S. Halevi, Y. Ishai, E. Kushilevitz, N. Makriyannis, and T. Rabin. On fully secure MPC with solitary output. In *Proceedings of the 17th Theory of Cryptography Conference(TCC), part I*, pages 312–340, 2019.

[28] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 230–235. IEEE Computer Society, 1989.

[29] Y. Lindell and T. Rabin. Secure two-party computation with fairness - A necessary design principle. In *Proceedings of the 15th Theory of Cryptography Conference(TCC), part I*, pages 565–580, 2017.

[30] H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, and A. Hithnawi. Rofl: Robustness of secure federated learning. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 453–476, 2023.

[31] N. Makriyannis. On the classification of finite Boolean functions up to fairness. In *Proceedings of the 9th Conference on Security and Cryptography for Networks (SCN)*, pages 135–154, 2014.

[32] N. Makriyannis. *Fairness in two-party computation: characterizing fair functions*. PhD thesis, Universitat Pompeu Fabra, 2016.

[33] T. Moran, M. Naor, and G. Segev. An optimally fair coin toss. In O. Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[34] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–85, 1989.

A Missing Proofs

A.1 Proof of Lemma 6.4

We now provide the proof of Lemma 6.4. The proof is nearly identical to the proof of [7, Lemma 4.2], and is taken almost verbatim. We first restate the lemma.

Lemma A.1 (Restatement of Lemma 6.4)**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a strong-semi balanced solitary output Boolean three-party functionality. Assume that there exists an r-round protocol for computing f with full security. Then, there exists a constant $\xi > 0$ (independent of the protocol) such that one of the following holds.*

- *There exists a real-world adversary corrupting* A *and* C *that can guess* $\mathsf{flip}(y)$ *with probability at least $\delta_2 + q^- + \xi/r$.*
- *There exists a real-world adversary corrupting* B *and* C *that can guess* $\mathsf{flip}(x)$ *with probability at least $\delta_1 + p^- + \xi/r$.*

Proof. Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a strong semi-balanced solitary output Boolean three-party functionality with normalized locking strategies $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ and $\mathbf{q} \in \mathbb{R}^{|\mathcal{Y}|}$ satisfying

$$\begin{cases} \mathbf{p}^T \cdot \mathbf{M}_z = \delta_1 \cdot \mathbf{1}^T, \text{ where } \delta_1 > 0 \\ \mathbf{1}^T \cdot \mathbf{p} < \delta_1, \\ \sum_{x \in \mathcal{X}} |p_x| = 1, \\ -\delta_1 + \mathbf{1}^T \cdot \mathbf{p} \leq \mathbf{p}^T \cdot \mathbf{M}_{z'}(\cdot, y) \leq \delta_1 \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{M}_z \cdot \mathbf{q} = \delta_2 \cdot \mathbf{1}, \text{ where } \delta_2 > 0 \\ \mathbf{1}^T \cdot \mathbf{q} < \delta_2, \\ \sum_{y \in \mathcal{Y}} |q_y| = 1, \\ -\delta_2 + \mathbf{1}^T \cdot \mathbf{q} \leq \mathbf{M}_{z'}(x, \cdot) \cdot \mathbf{q} \leq \delta_2 \end{cases},$$

for some $z \in \mathcal{Z}$, $z' \in \mathcal{Z}$, $\delta_1 = (\sum_{x \in \mathcal{X}} |\widetilde{p}_x|)^{-1}$, and $\delta_2 = (\sum_{y \in \mathcal{Y}} |\widetilde{q}_y|)^{-1}$.

Let $\mathsf{flip}(x)$ output 1 if $p_x < 0$, and 0 otherwise. Similarly, let $\mathsf{flip}(y)$ output 1 if $q_y < 0$, and 0 otherwise. Let $p^- = \sum_{x \in \mathcal{X} : p_x < 0} |p_x|$ denote the probability that $\mathsf{flip}(x) = 1$ and let $q^- = \sum_{y \in \mathcal{Y} : q_y < 0} |q_y|$ denote the probability that $\mathsf{flip}(y) = 1$. We further let $p^+ = 1 - p^-$ and $q^+ = 1 - q^-$. For all $i \in \{0, \ldots, r\}$ we let $\widetilde{a}_i = a_i \oplus \mathsf{flip}(x)$. Similarity, we let $\widetilde{b}_i = b_i \oplus \mathsf{flip}(y)$.

We will use the following properties proved by [31] that any *2-ary* strong semi-balanced functionality satisfies. Note that since we fixed $z \in \mathcal{Z}$, we can think of the functionality $f(x, y, z)$ as the 2-ary functionality $f_z(x, y) = f(x, y, z)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

Lemma A.2 ([7, Lemma 4.4])**.** *Let $f : \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \to \{0,1\}$ be a strong-semi balanced solitary output Boolean three-party functionality, with normalized z-locking strategies $\mathbf{p} \in \mathbb{R}^{|\mathcal{X}|}$ and $\mathbf{q} \in \mathbb{R}^{|\mathcal{Y}|}$ for A, and B, respectively, for some $z \in \mathcal{Z}$. Define $\mathsf{flip}(\cdot)$, p^+, p^-, q^+, q^-, δ_1, and δ_2 as before. Then, the following hold.*

1. *[31, Lemma 6.4:] $(p^+ - p^-)\delta_2 = (q^+ - q^-)\delta_1$.*
2. *[31, Lemma 6.5:] For all $y \in \mathcal{Y}$ it holds that $\Pr_{x \leftarrow |\mathbf{p}|}[f(x, y, z) \oplus \mathsf{flip}(x) = 1] = \delta_1 + p^-$.*
3. *[31, Lemma 6.5:] For all $x \in \mathcal{X}$ it holds that $\Pr_{y \leftarrow |\mathbf{q}|}[f(x, y, z) \oplus \mathsf{flip}(y) = 1] = \delta_2 + q^-$.*

First, we show that the distributions of every $\widetilde{a}_i$ and $\widetilde{b}_i$ are fixed throughout the execution of π.

Claim A.3 ([7, Claim 4.5])**.** *For every $i \in \{0, \ldots, r\}$ it holds that*

$$\left|\Pr[\widetilde{a}_i = 1] - (\delta_1 + p^-)\right| = \mathsf{neg}(\kappa) \quad \textit{and that} \quad \left|\Pr\left[\widetilde{b}_i = 1\right] - (\delta_2 + q^-)\right| = \mathsf{neg}(\kappa)$$

Proof. We prove the second assertion (the first one is analogous). For that, fix $i \in \{0, \ldots, r\}$ and fix a real-world adversary $\mathcal{A}$ that corrupts only A and instructs it to abort after receiving i messages from B. The output of an honest C in this case is b_i. By Item 3 of Lemma A.2 for all $x \in \mathcal{X}$ it holds that $\Pr_{y \leftarrow |\mathbf{q}|}[f(x, y, z) \oplus \mathsf{flip}(y) = 1] = \delta_2 + q^-$. Since π is assumed to be secure, there exists an ideal world simulator Sim for $\mathcal{A}$. Denote by w the output of the honest party in the ideal world. By Item 3 of Lemma A.2 it holds that $\Pr[w \oplus \mathsf{flip}(y) = 1] = \delta_2 + q^-$, regardless of what Sim sends to the trusted party. Hence, up to a negligible difference, the same holds in the real world. □

Note that as $\delta_1 > \mathbf{p}^T \cdot \mathbf{1} = p^+ - p^-$, it holds that $\delta_1 + p^- \geq -\delta_1 + p^+$. Similarly, it holds that $\delta_2 + q^- \geq -\delta_2 + q^+$. The next claim asserts that at some round i, there is a "jump" in the distribution of the (possibly flipped) backup values.

Claim A.4 ([7, Claim 4.6])**.** *There exists a value $w \in \{0,1\}$ and a constant $\xi > 0$ (independent of the protocol), such that one of the following holds. There exists a round $i \in [r]$ such that either*

$$\left|\Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i \neq w \wedge \widetilde{b}_i = 1\right] - (\delta_2 + q^-)\right| \geq \xi/r,$$

or

$$\left|\Pr\left[\widetilde{b}_{i-1} = w \wedge \widetilde{a}_{i-1} = 1\right] + \Pr\left[\widetilde{b}_i \neq w \wedge \widetilde{a}_i = 1\right] - (\delta_1 + p^-)\right| \geq \xi/r.$$

Proof. According to Claim A.3, for all $i \in [r]$ and all $w \in \{0,1\}$ it holds that

$$\begin{aligned}
&\Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i \neq w \wedge \widetilde{b}_i = 1\right] - (\delta_2 + q^-) \\
&\geq \Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i \neq w \wedge \widetilde{b}_i = 1\right] - \Pr\left[\widetilde{b} = 1\right] - \mathsf{neg}(\kappa) \\
&= \Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] - \Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa).
\end{aligned}$$

Similarly, it holds that

$$\begin{aligned}
&\Pr\left[\widetilde{b}_{i-1} = w \wedge \widetilde{a}_{i-1} = 1\right] + \Pr\left[\widetilde{b}_{i-1} \neq w \wedge \widetilde{b}_i = 1\right] - (\delta_1 + p^-) \\
&\geq \Pr\left[\widetilde{b}_{i-1} = w \wedge \widetilde{a}_{i-1} = 1\right] + \Pr\left[\widetilde{b}_{i-1} = w \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa).
\end{aligned}$$

Denote by Δ the average of the absolute values of the above taken over all $i \in [r]$ and $w \in \{0,1\}$. That is,

$$\begin{aligned}
\Delta := \frac{1}{4r} \sum_{i=1}^{r} \sum_{w=0}^{1} \Bigg[&\left|\Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] - \Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa)\right| \\
&+ \left|\Pr\left[\widetilde{b}_{i-1} = w \wedge \widetilde{a}_{i-1} = 1\right] + \Pr\left[\widetilde{b}_{i-1} = w \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa)\right| \Bigg].
\end{aligned}$$

Multiplying by $4r$ and separating the inner sum results in

$$\begin{aligned}
4r \cdot \Delta = \sum_{i=1}^{r} \Bigg[& \left|\Pr\left[\widetilde{a}_i = 0 \wedge \widetilde{b}_{i-1} = 1\right] - \Pr\left[\widetilde{a}_i = 0 \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa)\right| \\
& + \left|\Pr\left[\widetilde{b}_{i-1} = 0 \wedge \widetilde{a}_i = 1\right] - \Pr\left[\widetilde{b}_{i-1} = 0 \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa)\right| \\
& + \left|\Pr\left[\widetilde{a}_i = 1 \wedge \widetilde{b}_{i-1} = 1\right] - \Pr\left[\widetilde{a}_i = 1 \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa)\right| \\
& + \left|\Pr\left[\widetilde{b}_{i-1} = 1 \wedge \widetilde{a}_{i-1} = 1\right] - \Pr\left[\widetilde{b}_{i-1} = 1 \wedge \widetilde{b}_i = 1\right] - \mathsf{neg}(\kappa)\right| \Bigg] \\
= \sum_{i=1}^{r} \Bigg[& \Pr\left[\widetilde{a}_i = 0 \wedge \widetilde{b}_{i-1} = 0\right] - \Pr\left[\widetilde{a}_i = 0 \wedge \widetilde{b}_i = 0\right] + \mathsf{neg}(\kappa) \\
& + \Pr\left[\widetilde{b}_{i-1} = 0 \wedge \widetilde{a}_i = 0\right] - \Pr\left[\widetilde{b}_{i-1} = 0 \wedge \widetilde{b}_i = 0\right] + \mathsf{neg}(\kappa) \\
& + \Pr\left[\widetilde{a}_i = 1 \wedge \widetilde{b}_{i-1} = 1\right] - \Pr\left[\widetilde{a}_i = 1 \wedge \widetilde{b}_i = 1\right] + \mathsf{neg}(\kappa) \\
& + \Pr\left[\widetilde{b}_{i-1} = 1 \wedge \widetilde{a}_{i-1} = 1\right] - \Pr\left[\widetilde{b}_{i-1} = 1 \wedge \widetilde{b}_i = 1\right] + \mathsf{neg}(\kappa) \Bigg] \\
\geq \sum_{i=1}^{r} \Bigg[& \Pr\left[\widetilde{a}_i = \widetilde{b}_{i-1}\right] - \Pr\left[\widetilde{a}_i = \widetilde{b}_i\right] + 2\mathsf{neg}(\kappa) \\
& + \Pr\left[\widetilde{b}_{i-1} = \widetilde{a}_i\right] - \Pr\left[\widetilde{b}_{i-1} = \widetilde{b}_i\right] + 2\mathsf{neg}(\kappa) \Bigg] \\
\geq \Pr\left[\widetilde{a}_0 = \widetilde{b}_0\right] & - \Pr\left[\widetilde{a}_r = \widetilde{b}_r\right] - 2r \cdot \mathsf{neg}(\kappa),
\end{aligned}$$

where the inequalities follow from the triangle inequality.

Since $\widetilde{a}_0$ and $\widetilde{b}_0$ are computed before any interaction is made, they are independent. Thus, by Claim A.3 it holds that

$$\left|\Pr\left[\widetilde{a}_0 = \widetilde{b}_0\right] - \Big((\delta_1 + p^-)(\delta_2 + q^-) + (-\delta_1 + p^+)(-\delta_2 + q^+)\Big)\right| = \mathsf{neg}(\kappa).$$

Moreover, since $\widetilde{a}_r$ and $\widetilde{b}_r$ correspond to the possibly flipped output of the protocol, it holds that they are equal if and only if $\mathsf{flip}(x) = \mathsf{flip}(y)$. Thus, it follows that

$$\left|\Pr\left[\widetilde{a}_r = \widetilde{b}_r\right] - (p^- q^- + p^+ q^+)\right| = \mathsf{neg}(\kappa).$$

We get that

$$\Delta \geq \frac{1}{4r}\left|2\delta_1\delta_2 + (q^- - q^+)\delta_1 + (p^- - p^+)\delta_2\right| - \mathsf{neg}(\kappa).$$

By Item 1 of Lemma A.2 it holds that $(q^- - q^+)\delta_1 = (p^- - p^+)\delta_2$. Thus, it holds that

$$\Delta \geq \frac{\delta_1}{2r}\left|\delta_2 + q^- - q^+\right| - \mathsf{neg}(\kappa).$$

Since $\delta_1 \neq 0$ and $\delta_2 \neq \mathbf{q}^T \cdot \mathbf{1} = q^+ - q^-$, it follows that for $\xi := \delta_1 \cdot |\delta_2 + q^- - q^+|/3 > 0$ it holds that

$$\Delta \geq \frac{\delta_1}{2r}|\delta_2 + q^- - q^+| - \mathsf{neg}(\kappa) \geq \xi/r.$$

Thus, by averaging argument the claim follows. □

We are now ready to construct the adversary. Assume without loss of generality that there exists a round $i \in [r]$ for which it holds that

$$\Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i \neq w \wedge \widetilde{b}_i = 1\right] - (\delta_2 + q^-) \geq \xi/r.$$

The case where

$$\Pr\left[\widetilde{a}_i = w \wedge \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i \neq w \wedge \widetilde{b}_i = 1\right] - (\delta_2 + q^-) \leq -\xi/r$$

can be handled by observing that

$$\begin{aligned}
&\Pr\left[\widetilde{a}_i = w, \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i \neq w, \widetilde{b}_i = 1\right] - (\delta_2 + q^-) \\
&= \Pr\left[\widetilde{a}_i = w\right] - \Pr\left[\widetilde{a}_i = w, \widetilde{b}_{i-1} = 0\right] + \Pr\left[\widetilde{a}_i \neq w\right] - \Pr\left[\widetilde{a}_i \neq w, \widetilde{b}_i = 0\right] - (\delta_2 + q^-) \\
&= -\left(\Pr\left[\widetilde{a}_i = w, \widetilde{b}_{i-1} = 0\right] + \Pr\left[\widetilde{a}_i \neq w, \widetilde{b}_i = 0\right] - (1 - \delta_2 - q^-)\right),
\end{aligned}$$

and then applying an analogous argument.

We define the adversary $\mathcal{A}$ that corrupts both A and C as follows.

1. Corrupt both A and C, instruct C to set z as its input, and instruct both of them to act honestly until receiving i messages from B.
2. Compute the backup value a_i as an honest A and C would in case B aborts, and do the following.
   (a) If $a_i \oplus \mathsf{flip}(x) = 1$, then instruct A to send its next message honestly and then abort.
   (b) If $a_i \oplus \mathsf{flip}(x) = 0$, then instruct A to abort immediately.
   (c) In both cases, C acts honestly until the end of the execution, where it obtains the output b.
3. Output $b \oplus 1$ as the guess for $\mathsf{flip}(y)$.

Observe that the following holds.

$$\begin{aligned}
\Pr\left[b \oplus 1 = \mathsf{flip}(y)\right] &= \Pr\left[\widetilde{a}_i = 1 \wedge b_{i-1} \oplus 1 = \mathsf{flip}(y)\right] + \Pr\left[\widetilde{a}_i = 0 \wedge b_i \oplus 1 = \mathsf{flip}(y)\right] \\
&= \Pr\left[\widetilde{a}_i = 1, \widetilde{b}_{i-1} = 1\right] + \Pr\left[\widetilde{a}_i = 0, \widetilde{b}_i = 1\right] \\
&\geq \delta_2 + q^- + \xi/r,
\end{aligned}$$

which completes the proof. □

A.2 Dealing with a Corrupted B in the Solitary Output Setting

To complete the proof of Lemma 4.4, we need to simulate adversaries that either corrupt B, or corrupt B and C. We first handle the case where only B is corrupted. Let $\mathcal{B}$ be a real-world adversary that corrupts B. We define the simulator as follows.

1. Query the adversary for the input y it sends to the dealer.
2. Sample $i^* \leftarrow \mathrm{Geom}(\alpha)$ according to the geometric distribution with parameter α.
3. For $i = 1$ to $i^* - 1$:
   (a) Send to $\mathcal{B}$ a random bit which represents its share of the backup value b_i.
   (b) If $\mathcal{B}$ aborts B, then sample $y^* \leftarrow \mathcal{Y}$, send y^* to the trusted party, and output random shares as the view of the adversary and halt.
4. Send y to the trusted party and receive $w = f(x, y, z)$.
5. For $i = i^* + 1$ to r:
   - Send to $\mathcal{B}$ a random bit which represent its share from w.
   - If $\mathcal{B}$ aborts B then output random shares as the view of the adversary and halt.
6. output random shares as the view of the adversary and halt.

Let i denote the round where the adversary instructs B to abort (if no such round exists, set $i = r + 1$). First, note that in both worlds the view of B consists only of random and independent shares. Thus, it suffices to show that the output distribution of C is the same in both worlds. If $i < i^*$ then the output of C in both worlds is $f(x, \widetilde{y}, z)$, for $y \leftarrow \mathcal{Y}$. If $i \geq i^*$ the output of C part is $f(x, y, z)$. In both cases, the output of C is identically distributed in both worlds.

We now consider an adversary $\mathcal{B}$ that corrupts both B and C. First, recall that in any protocol in the dealer model, the adversary cannot abort C. We next describe the simulator.

1. Query the adversary for the inputs y and z it sends to the dealer.
2. Sample random shares $(a_i[\mathsf{C}], b_i[\mathsf{C}])_{i=0}^{r}$ and send them to $\mathcal{B}$.
3. Sample $i^* \leftarrow \mathrm{Geom}(\alpha)$ according to the geometric distribution with parameter α.
4. For $i = 1$ to $i^* - 1$:
   (a) If $i < i^* - 1$, then compute $b_i = f(\widetilde{x}_i, y, z)$, where $\widetilde{x}_i \leftarrow \mathcal{X}$. Otherwise, set $b_i = 1$ with probability $\beta_{y,z}$ and $b_i = 0$ with probability $1 - \beta_{y,z}$.
   (b) Send $b_i[\mathsf{B}] := b_i - b_i[\mathsf{C}]$ to $\mathcal{B}$.
   (c) If $\mathcal{B}$ aborts B then do the following.
      i. Sample $y^* \leftarrow \mathcal{Y}$, send (y^*, z) to the trusted party and receive $w = f(x, y^*, z)$.
      ii. Send $a_i[\mathsf{A}] := w - a_i[\mathsf{C}]$ to the adversary, output whatever it outputs, and halt.
5. Send (y, z) to the trusted party and receive $w = f(x, y, z)$.
6. For $i = i^*$ to r:
   (a) Send $b_i[\mathsf{B}] = w - b_i[\mathsf{C}]$ to $\mathcal{B}$.
   (b) If $\mathcal{B}$ aborts B then send it $a_i[\mathsf{A}] := w - a_i[\mathsf{C}]$, output whatever it outputs, and halt.
7. Output whatever $\mathcal{B}$ outputs and halt.

Denote by i the round where the adversary instructs B to abort (if no such round exists, set to $r+1$). Observe that in the real world, the adversary holds enough shares to reconstruct the backup values $b_1, \ldots, b_i$ and a_i. If $i > i^*$, then the view of the adversary in both worlds is equal to $b_1, \ldots, b_i$ and a_i, where $a_i = b_i = f(x, y, z)$. If $i = i^* - 1$, then the view of then the view of the adversary in both worlds is equal to $b_1, \ldots, b_i$ and a_i, where $a_i = f(x, \widetilde{y}, z)$ for $\widetilde{y} \leftarrow \mathcal{Y}$, and $b_i = 1$ with probability $\beta_{y,z}$ and $b_i = 0$ with probability $1 - \beta_{y,z}$. Finally, if $i < i^* - 1$, then the view of the adversary in both worlds is equal to $b_1, \ldots, b_i$ and a_i, where $a_i = f(x, \widetilde{y}, z)$ for $\widetilde{y} \leftarrow \mathcal{X}$, and $b_i = f(\widetilde{x}, y, z)$ for $\widetilde{x} \leftarrow \mathcal{Y}$. Also note that if $i < i^*$, then all the backup values $b_1, \ldots, b_i$ and a_i are independent in both worlds. Thus, the joint distribution of (a_i, b_i) is identically distributed in both worlds.

A.3 Dealing with a Corrupted B in the Two-Party Setting

To complete the proof of Lemma 5.10, we need to simulate any adversary corrupting B. Let $\mathcal{B}$ be such a real-world adversary. We define its simulator as follows.

1. Query the adversary for the input y it sends to the dealer.
2. Sample $i^* \leftarrow \mathrm{Geom}(\alpha)$ according to the geometric distribution with parameter α.
3. For $i = 1$ to $i^* - 1$:
   (a) Compute $b_i = f(\widetilde{x}_i, y)$, where $\widetilde{x}_i \leftarrow \mathcal{X}$, and send it to $\mathcal{B}$.
   (b) If $\mathcal{B}$ aborts B, then sample $y^* \leftarrow \mathcal{Y}$ and send y^* to the trusted party and halt.
4. Send y to the trusted party and receive $w = f(x, y)$.
5. For $i = i^*$ to r:
   - Send $b_i = w$ to $\mathcal{B}$.
   - If $\mathcal{B}$ aborts B, then output whatever it outputs and halt.
6. Output whatever $\mathcal{B}$ outputs and halt.

Let i denote the round where the adversary instructs B to abort (if no such round exists, set $i = r + 1$). Observe that if $i < i^*$ then the output of A is $f(x, \widetilde{y})$ for $\widetilde{y} \leftarrow \mathcal{Y}$ in both worlds. If $i \geq i^*$ then the output of the honest party is $f(x, y)$. In both cases, the view of $\mathcal{B}$ is identically distributed given the output. Thus, the real and ideal worlds are identically distributed.