Statistical Security in Two-Party Computation Revisited

Saikrishna Badrinarayanan*
Snap

Sikhar Patranabis†
IBM Research India

Pratik Sarkar‡
Boston University

Abstract

We present a new framework for building round-optimal one-sided statistically secure two party computation (2PC) protocols in the plain model. We demonstrate that a relatively weak notion of oblivious transfer (OT), namely a three round elementary oblivious transfer eOT with statistical receiver privacy, along with a non-interactive commitment scheme suffices to build a one-sided statistically secure two party computation protocol with black-box simulation. Our framework enables the first instantiations of round-optimal one-sided statistically secure 2PC protocols from the CDH assumption and certain families of isogeny-based assumptions.

As part of our compiler, we introduce the following new one-sided statistically secure primitives in the pre-processing model that might also be of independent interest:

1. Three round statistically sender private random-OT where only the last OT message depends on the receiver's choice bit and the sender receives random outputs generated by the protocol.
2. Four round delayed-input statistically sender private conditional disclosure of secrets where the first two rounds of the protocol are independent of the inputs of the parties.

The above primitives are directly constructed from eOT and hence we obtain their instantiations from the same set of assumptions as our 2PC.

*Work done while the author was affiliated with Visa Research USA.

†Most of the work was done while the author was affiliated with Visa Research USA.

‡Supported by NSF Awards 1931714, 1414119, and the DARPA SIEVE program.

Contents

1 Introduction

Secure two party computation (2PC) enables two mutually distrusting parties to compute a function on their private inputs without revealing anything beyond their output. An important question in the study of secure computation has been designing protocols in minimal rounds. The phenomenal work of Katz and Ostrovsky [KO04] showed that four rounds are necessary when one party receives the output and five rounds are necessary if both parties wish to receive output. Starting with [KO04], there has been a large body of work in designing round-optimal protocols in the plain model, secure against a probabilistic polynomial time (PPT) malicious adversary, in the two-party setting [ORS15, COSV17, CCG+21] and the multi-party setting with dishonest majority [GMPP16, BHP17, ACJ17, BGJ+18, HHPV18, CCG+20].

Statistical Security. A natural question to ask is can we obtain round optimal protocols when the parties are computationally unbounded? For the specific problem of zero knowledge proofs/arguments, this question has been well studied [GMW91, Nao91, GK96, BJY97, NOVY98, HNO+09]. In particular, assuming collision resistant hash functions: (i) Statistical zero knowledge arguments for NP, where soundness is computational and zero knowledge is statistical, are known in four rounds (round optimal) with black-box simulation [BJY97] and (ii) Computational zero knowledge proofs for NP, that satisfy statistical soundness and computational zero knowledge, are known in five rounds (round optimal) with black-box simulation [GK96]. There has also been work on building round-optimal (two rounds) statistically secure protocols for weaker functionalities like ZAPs and witness indistinguishable proofs/arguments [DN07, KKS18, BFJ+20, GJJM20].

Handling computationally unbounded adversaries for general two party functionalities is more challenging. For instance, Katz [Kat08] proved that it is impossible to obtain four round zero knowledge (ZK) proofs. This immediately rules out statistical security in four rounds for a two party secure computation protocol where only one party (denoted as the receiver) wishes to learn the output and the other party (denoted as the sender) is computationally unbounded. Therefore, the best possible security that one can hope for in four rounds is security against a computationally unbounded receiver and a PPT sender. This was termed as *one-sided statistical security* by Khurana and Mughees [KM20]. The works of [OPP14, CO17, KKS18] considered weaker notions such as one-sided statistical security with respect to super-polynomial time simulation. However, the question of obtaining one-sided statistically secure protocols with (standard) polynomial-time black-box simulation remained elusive for a long time. Only recently, this question was addressed by the work of [KM20]. They constructed round-optimal one-sided statistically secure two-party computation protocols with black-box simulation-based security against malicious adversaries:

- A four round statistically sender private (SSP) protocol where the receiver obtains the output at the end of fourth round,
- A five round statistically sender private protocol where the receiver obtains the output at the end of fourth round and the sender obtains the output at the end of fifth round.

The underlying building blocks in [KM20] are two-round statistically sender private OT ($\mathsf{SSP}_{\mathsf{OT}}$) [BD18, NP01, HK12] and a non-interactive commitment scheme. They instantiate

the above protocol based on Learning with Errors (LWE), Decisional Diffie Hellman (DDH) or Quadratic Residuosity (QR). However, it was left as an open problem in their work to study the minimal assumptions required to obtain round-optimal 2PC protocols with one-sided statistical security, following similar investigations on assumptions versus round complexity in zero knowledge arguments/proofs with statistical security. For instance, it is unknown whether we can build round-optimal one-sided statistically secure 2PC protocols from other standard assumptions such as the Computational Diffie-Hellman (CDH) or the newer class of isogeny-based assumptions. In this work, we ask the following question:

Can we construct round-optimal one-sided statistically secure 2PC protocols with black-box simulation in the plain model from a wider class of assumptions?

1.1 Our Contributions

We answer the above question in the affirmative. We establish a general compiler to achieve round-optimal one-sided statistically secure 2PC protocols that relies on potentially weaker (or "less structured") cryptographic primitives as compared to those used by [KM20]. These primitives can be instantiated from essentially *all* commonly used cryptographic assumptions, including new instantiations from the CDH assumption and certain isogeny-based assumptions such as the Reciprocal CSIDH assumption (which were not known before and are contributions of this work), as well as instantiations from LWE, LPN (+ derandomization techniques)[1], Quadratic Residuosity, N^{th} Residuosity, and decisional CSIDH (all of which follow from existing works). In particular, the new instantiations from CDH and Reciprocal CSIDH are enabled precisely by the usage of potentially weaker (or "less structured") cryptographic primitives in our framework as compared to those used by [KM20]. Our approach is conceptually similar to that taken by the authors of [AMPS21] to weaken the underlying primitives for round-optimal secure computation (MPC) protocols which are secure against adaptive corruption of parties, but the techniques used by our compiler are fundamentally different.

Our Ingredients. We introduce the notion of statistically receiver private (SRP) elementary OT in the plain model following the work of Dottling et al. [DGH+20]. We denote it as eOT[2] throughout the paper. It is a three round OT protocol, where the sender, with no input, sends the first message that can be viewed as a pre-processing phase, the receiver sends the second message based on its choice bit, and then, the sender computes random outputs (which can be viewed as its two input messages in the traditional OT definition) and sends the final OT message. Elementary security ensures that a maliciously corrupt receiver is unable to compute both sender outputs. Statistical receiver privacy implies that the choice bit is statistically hidden from a maliciously corrupt sender, with unbounded computational power. We show that such an OT protocol combined with a non-interactive commitment scheme suffices for one-sided statistical security. This yields a four-round 2PC

[1] Throughout this paper, when we refer to the LPN assumption, we refer to the "extremely low-noise" variant of LPN with noise parameters in the $O\left((\log n)^2/n\right)$ regime, as used in many recent works, including [BF22].

[2] We consider that our eOT protocol provides statistical receiver privacy, as opposed to the elementary OT protocol defined in [DGH+20] which only provides computational receiver privacy.

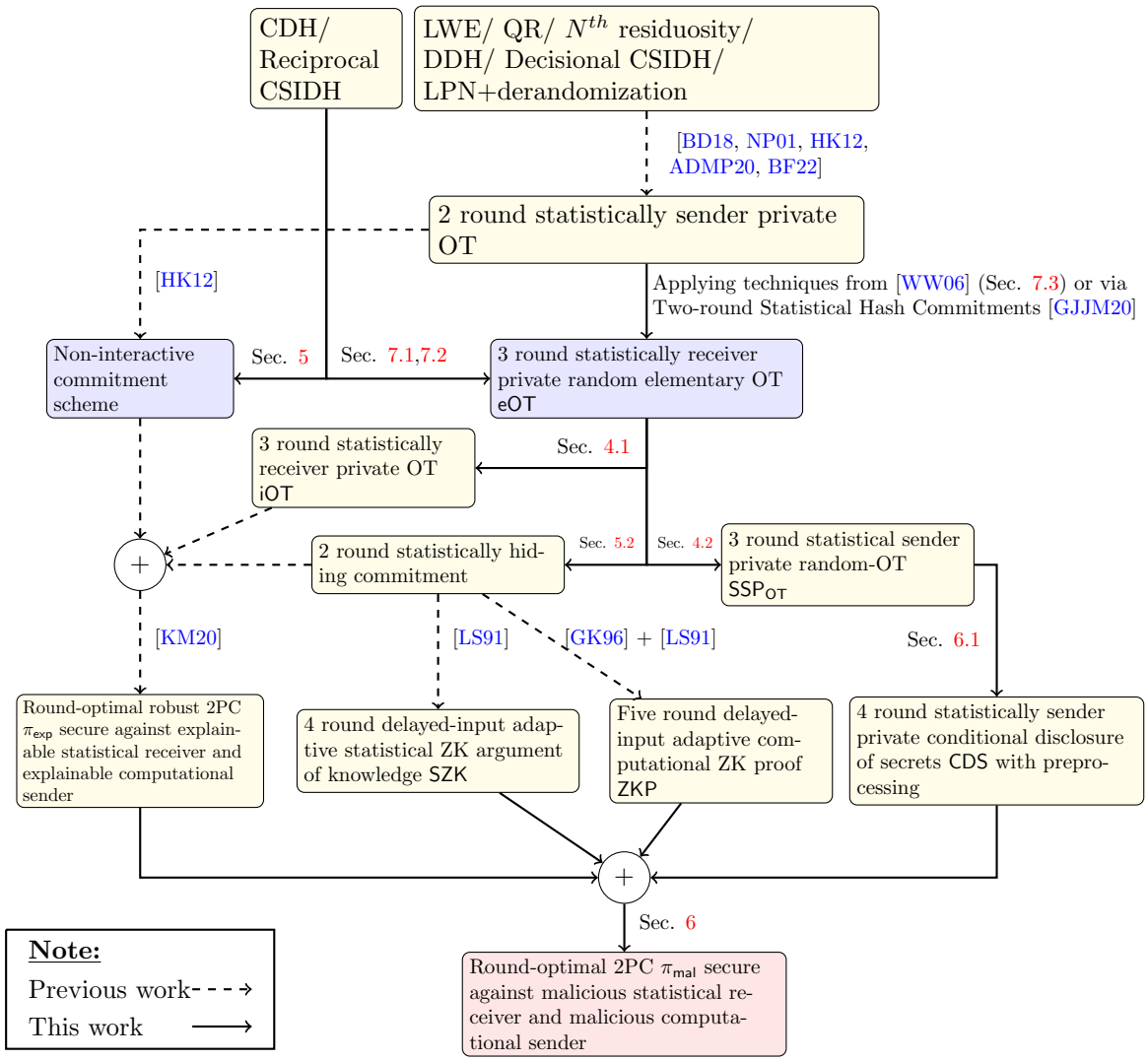

Figure 1: Roadmap of our compiler

protocol where the receiver obtains the output at the end of the fourth round, and a five-round protocol where both parties obtain the output. Our contributions are summarized in Thm. 1 and has been depicted in Fig. 1.

Theorem 1 (Informal) *Assuming a non-interactive commitment scheme and a three round statistically-receiver private elementary OT, denoted as* eOT*, there exists:*

- *A four-round 2PC protocol where the receiver obtains the output at the end of the protocol,*
- *A five-round 2PC protocol where both parties obtain the output.*

Both our protocols achieve statistical security against a malicious receiver and computational security against a malicious sender in the plain model[3] *and require black-box simulation.*

We demonstrate that a two-round $\mathsf{SSP}_{\mathsf{OT}}$ implies eOT and a non-interactive commitment scheme, hence *re-obtaining* the results of [KM20] through our compiler. Instantiating the $\mathsf{SSP}_{\mathsf{OT}}$ from LPN+Nissan Wigderson style derandomization [BF22] and isogeny-based assumption, like decisional CSIDH [ADMP20], we obtain new instantiations of our compiler. In addition, we also build eOT and non-interactive commitments from CDH and other isogeny based assumptions like reciprocal[4] CSIDH [LGdSG21]. This gives us one-sided statistical 2PC from CDH and reciprocal CSIDH which was not known before. Combining the above results, we obtain one-sided statistical 2PC from most well-studied assumptions in cryptography.

Theorem 2 (Informal) *Assuming CDH, LWE, LPN (+derandomization techniques), QR, N^{th} Residuosity, or isogeny-based assumptions (decisional CSIDH or Reciprocal CSIDH), there exists:*

- *A four-round 2PC protocol where the receiver obtains the output at the end of the protocol,*
- *A five-round 2PC protocol where both parties obtain the output.*

Both our protocols achieve statistical security against a malicious receiver and computational security against a malicious sender in the plain model and require black-box simulation.

As part of our building blocks, we introduce the notion of statistically sender private conditional disclosure of secrets CDS in the preprocessing model and demonstrate that eOT (and information theoretic garbling for NC1 circuits) suffices for its construction. This is a weakening of two-round statistically sender private conditional disclosure of secrets which is built from two-round $\mathsf{SSP}_{\mathsf{OT}}$. Our primitive could be of independent interest, especially in constructing one-sided statistically secure MPC protocols from different assumptions. Formally,

Theorem 3 *Assuming a three round statistically-receiver private elementary OT, there exists a four-round statistically sender private conditional disclosure of secrets CDS for NC1 circuits in the pre-processing model, where the first two rounds of CDS are input-independent.*

Instantiating eOT from the above assumptions, we obtain the CDS from most well-studied assumptions as follows.

Theorem 4 *Assuming CDH, LWE, LPN (+derandomization techniques), QR, N^{th} Residuosity, or isogeny-based assumptions (decisional CSIDH or Reciprocal CSIDH), there exists*

[3] For the five round protocol, receiver is the party that obtains output first (at the end of round four) and sender is the party that obtains output at the end of round five.

[4] Reciprocal CSIDH is quantum equivalent to computational CSIDH, which is weaker than decisional CSIDH. However, reciprocal CSIDH and decisional CSIDH assumptions are incomparable in the classical setting.

a four-round statistically sender private conditional disclosure of secrets CDS *for NC1 circuits in the pre-processing model, where the first two rounds of* CDS *are input-independent.*

The information theoretic garbling [Kol05] for NC1 circuits is used to construct the above CDS for NC1 circuits. The above CDS for NC1 circuits suffice for one-sided statistical 2PC for all circuits.

Roadmap. We provide a detailed overview of our protocols in Sec. 2. Then we define our building blocks in Sec. 3. We define other OT protocols in Sec. 4 and construct them from eOT. These OTs protocol would be instrumental in our final compiler. We construct our round optimal one-sided statistically secure 2PC protocol π_{exp} against explainable parties in Sec. 5. Finally, we compile π_{exp} to obtain a round optimal one-sided statistically secure 2PC protocol π_{mal} which is secure against malicious corruptions in Sec. 6. In the same section we construct statistically sender private CDS in preprocessing model. Finally, we provide instantiations of eOT from different assumptions in Sec. 7.

2 Technical Overview

In this section we demonstrate that a three round statistically receiver private elementary OT, denoted as eOT, and a non-interactive commitment scheme suffices to obtain a five round (which is round optimal) 2PC protocol that obtains security against a computationally unbounded receiver and a PPT sender. Then we instantiate eOT and the commitment scheme from various assumptions. Along the way, we introduce new primitives of independent interest - statistical conditional disclosure of secrets CDS in the preprocessing model and a three round random SSP-OT, and instantiate them from various assumptions.

2.1 One-Sided Statistical Two-Party Computation Protocol

Our compiler builds upon the compiler of [KM20] by weakening the underlying primitives in their compiler. We recall their protocol for completeness. [KM20] constructed a five round 2PC protocol against malicious adversaries. The first party, denoted as the receiver, is computationally unbounded and obtains the output at the end of fourth round. The second party, called the sender, is computationally bounded and obtains the output at the end of fifth round. The protocol proceeds through two transformations- where [KM20] first constructs a protocol which is secure against explainable adversaries and then compiles it (interactive proofs) to obtain security against malicious adversaries.

Robust 2PC Secure against Explainable Adversaries. As the first step, [KM20] considered explainable adversaries[5] which generates protocol messages in the support of the distribution of all honestly generated transcripts, and the simulator needs to extract the input and randomness of the adversarial party from the transcript. In this setting, the

[5]It is different from the notion of semi-malicious security [MW16] where the adversary in addition to generating the the protocol messages in the support of the distribution of all honestly generated transcripts, also outputs the input and randomness that was used, on a special tape.

classical garbled circuit based approach of [Yao86], where the receiver is the evaluator and sender is the garbler, fails since the receiver is computationally unbounded and information theoretically private garbling scheme is known only for NC1 circuits.

Reversing the Roles. [KM20] takes a different approach where the receiver garbles the circuit and sender evaluates it. The sender obtains the wire labels corresponding to its input through a statistical receiver private OT, hence hiding its input against an unbounded corrupt receiver. The OT protocol takes three rounds, starting from the receiver (acting as the sender of the OT), and the garbled circuit is sent in the third round by the receiver.

Simulating against Explainable Parties. The simulator needs to simulate against explainable adversaries by extracting their inputs. To enable extraction of the corrupt sender's input the sender is also required to commit to its input using a four round statistically hiding and computationally binding extractable commitment scheme. Similarly, the receiver commits to its input and randomness using a three round statistically binding and computationally hiding extractable commitment scheme. These commitments allow a simulator to extract the input and randomness of the explainable adversarial parties. The simulator against a corrupt sender's extracts the sender's input at the end of fourth round from the commitment scheme and obtains the correct output only at the end of fourth round. However, the receiver is required to send the garbled circuit in the third round. This creates a problem in simulation since a corrupt sender, evaluating the garbled circuit, distinguishes an interaction with an honest receiver from an interaction with the simulated receiver based on the garbled circuit output.

One Last Modification. To avoid this, the receiver garbles a different circuit so that the garbled circuit computes an encryption of the output. The sender obtains the garbled circuit at the end of third round, evaluates it to obtain the encrypted output and then sends it to the receiver in the fourth round. The receiver decrypts the output and sends it to the sender in the fifth round. In the ideal world the simulator sends a simulated garbled circuit which outputs an encryption of 0 to the corrupt sender, hence providing correct simulation in the ideal world. [KM20] also ensured that the first two rounds of the protocol are robust - i.e. if the parties behave maliciously in the first 2 rounds of the protocol then they can influence the protocol output but they would fail to infer any information about the honest party's input. The robustness property is crucial when we upgrade to security against malicious adversaries.

Summary. To summarize the result of [KM20] they obtain a robust 5-round secure two-party computation protocol π_{exp} with black-box simulation against unbounded explainable receivers and PPT explainable senders, where the receiver obtains its output at the end of fourth round and the sender obtains its output at the end of the fifth round. Their underlying primitives are as follows:

1. Three round oblivious transfer with statistical privacy for a receiver and computational privacy for a sender,
2. Three round statistically binding and computationally hiding commitment scheme satisfying extractability.

3. Four round statistically hiding and computationally binding commitment scheme satisfying extractability.

4. Information theoretic garbled circuits for NC1 circuits (used by [KM20] to validate a specific NC1 relation as part of their statistically secure two-round CDS protocol – we expand more on this subsequently).

Overview of Our Contributions. We demonstrate that an elementary OT protocol (denoted simply as eOT in rest of the paper) and a non-interactive commitment scheme suffices to instantiate the above primitives and hence yield the protocol π_{exp} from eOT and a non-interactive commitment scheme.

1. The three round SRP-OT protocol, denoted as iOT, that satisfies indistinguishability based sender security is built from eOT in Sec. 4.1 in a round preserving manner. We discuss it in Sec. 2.2.

2. The three round statistically binding and computationally hiding commitment scheme can be constructed [PRS02] from any non-interactive commitment scheme.

3. The four round statistically hiding and computationally binding commitment scheme satisfying extractability can be obtained [KM20] by replacing the non-interactive commitment scheme in [PRS02] with a two round statistically hiding commitment scheme. We build the two round statistically hiding commitment scheme from SRP iOT in Sec. 5.2 and we briefly discus about it in Sec. 2.2.

4. Garbled circuits can be obtained [Yao86, LP07] from one way functions.

The Final Compiler. Next, the security of π_{exp} is uplifted such that it is secure against malicious adversaries using zero knowledge protocols as follows.

Tackling a Malicious Sender. The sender is required to prove that it generated the second and fourth round messages of π_{exp} correctly. This is performed using a four round delayed-input statistical zero knowledge protocol SZK where the input statement is chosen by the sender (behaving as the prover) in the last round of SZK. SZK is run in parallel to π_{exp} and the robustness of the first two rounds of π_{exp} ensures that the input of an honest receiver is not leaked even if a corrupt sender constructs the second round message of π_{exp} maliciously. We also require SZK to be an argument of knowledge for reasons, discussed later. SZK can be built [LS91] from two round statistically hiding commitment scheme.

Tackling a Malicious Receiver. Similar to the sender, the receiver is required to prove that it generated the first, third and fifth round messages of π_{exp} correctly. This is performed using a five round delayed-input zero knowledge proof ZKP where the input statement is chosen by the receiver (behaving as the prover) in the last round of ZKP. ZKP is run in parallel to π_{exp} and the robustness of the first two rounds of π_{exp} ensures that the input of an honest sender is not leaked even if a corrupt receiver constructs the first round message of π_{exp} maliciously. ZKP can be built ([LS91]+ [GK96]) from two round statistically hiding commitment scheme. However, a maliciously constructed third round message of π_{exp} could

leak an honest sender's input when the sender sends the fourth round message of π_{exp}. ZKP fails to address this issue since it takes five rounds to complete and an honest sender could detect the malicious behavior of a corrupt receiver only at the end of the fifth round. This leaks the honest sender's inputs.

Conditional Disclosure of Secrets. [KM20] addresses the above situation by using a two round conditional disclosure of secrets CDS where the receiver sends the public key for the CDS alongwith the third round message of π_{exp}. The sender encrypts the fourth round message of π_{exp} under the CDS public key and input statement - the first and third round message of π_{exp} is constructed in an explainable manner by the receiver. The receiver successfully decrypts the fourth round message of π_{exp} if it produces a witness attesting to the fact that the first and third round message of π_{exp} is explainable. If the receiver fails to produce such a witness, then the CDS plaintext (fourth round message of π_{exp}) remains statistically hidden. The CDS protocol requires soundness against a statistical receiver and witness privacy against a semi-honest computationally bounded sender. In the final protocol, the sender is required to prove in the fourth round that it constructed the CDS sender message correctly using SZK since the CDS provides security guarantees against a semi-honest sender. We require the SZK to be argument of knowledge so that the simulator (against a corrupt sender) is able to extract the encrypted CDS plaintext, which is the fourth round message of π_{exp}, in order to generate the final message of the protocol.

Note that [KM20] constructs an NC1 circuit which checks the validity of receiver's witness. The authors of [KM20] then proceed to construct a (two-round) CDS protocol with statistical security for the class of relations that are verifiable by NC1 circuits by combining two round statistically sender private OT with information-theoretic garbled circuits for NC1. In fact, it can be shown (via dashed lines in Fig. 1) that two-round statistically sender private OT suffices to instantiate the 2PC protocol of [KM20]. However, two-round statistically sender private OT is a relatively strong primitive and is not known from many well-studied assumptions, like CDH.

Our Proposal. We shift our starting point to presumably weaker primitives - a three round SRP eOT protocol where only the second OT message depends on the receiver's input and the sender's outputs are random. We show that eOT suffices for compiling π_{exp} to our final protocol π_{mal} which provides statistical security against a malicious receiver and computational security against a malicious sender as follows:

1. By applying round-preserving transformations on eOT we obtain a three round delayed-input statistical sender private OT protocol $\mathsf{SSP}_{\mathsf{OT}}$ - where only the last OT message depends on the receiver's input and the sender's outputs are random. Combining $\mathsf{SSP}_{\mathsf{OT}}$ with information theoretic garbling for NC1 we obtain a four round statistical CDS protocol where the first two rounds, aka preprocessing phase, are independent of the input statement and the witness. This new primitive suffices for conditional disclosure of secrets in the above 2PC protocol since the first two rounds can be used for the preprocessing phase of the $\mathsf{SSP}_{\mathsf{OT}}$ and the last two rounds can be used to run the input-dependent phase of CDS.
2. The four round delayed-input statistical zero knowledge SZK can be built [LS91] from two round statistically hiding commitment.

3. The five round delayed-input zero knowledge proof ZKP can be obtained [LS91]+ [GK96] from two round statistically hiding commitment.

4. The first two rounds of iOT implies a two round statistically hiding commitment scheme.

Previously, we have shown that π_{exp} can be obtained from a non-interactive commitment scheme and eOT. Combining the two results, we obtain our one-sided 2PC protocol from a non-interactive commitment scheme and eOT.

Instantiations. We demonstrate that a two-round statistical sender private OT implements eOT (by applying OT reversal techniques [WW06]) and the first message of the two-round statistical sender private OT is a non-interactive commitment scheme. Hence, our result generalizes the work of [KM20] and we obtain instantiations from LWE, QR, N^{th} residuosity, DDH, Decisional CSIDH and LPN+Nissan Wigderson derandomization by instantiating [BD18, NP01, HK12, HK12, ADMP20, BF22] the underlying two-round statistical sender private OT from the above assumptions. Furthermore, we build eOT and the non-interactive commitment scheme from CDH and reciprocal CSIDH [LGdSG21] assumptions. This was not previously known from [KM20]. To summarize, our proposed framework enables one-sided statistical 2PC from essentially all well-studied cryptographic assumptions.

2.2 Constructing our Ingredients from eOT

Next, we briefly introduce our ingredient primitives and discuss their constructions. A roadmap explaining our framework based on these ingredients can be found in Fig. 1.

Three Round Statistically Receiver Private eOT. We introduce the notion of statistically receiver private elementary OT in plain model following the work of [DGH+20]. It is a three round OT protocol where the sender sends the first message as a preprocessing phase, the receiver sends the second message based on its choice bit, and the sender sends the third message. The sender obtains random outputs. The elementary security ensures that a maliciously corrupt receiver is unable to compute both sender outputs. Statistical receiver privacy implies that the choice bit is statistically hidden from a maliciously corrupt sender. We show in Sec. 7.3 that a two round statistically sender private OT can be used to build eOT through OT reversal techniques [WW06], hence obtaining instantiations from a wide variety of assumptions (namely LWE, QR, N^{th} residuosity, DDH, Decisional CSIDH and LPN+Nissan Wigderson derandomization). We also construct eOT from CDH by building upon the two-round CDH based protocol of [DGH+20] in the crs model. In the CDH-based eOT instantiation, the sender sends the crs of the CDH based protocol of [DGH+20] as the OT first message and then their two-round CDH based protocol is run between the parties using the first message as the crs. The full construction can be found in Sec. 7.1. We also provide a new construction of eOT based on reciprocal CSIDH assumption in Sec. 7.2 which was previously not known. [6]

[6] Reciprocal CSIDH assumption is quantum equivalent to computational CSIDH and it is incomparable to decisional CSIDH,

Three Round Statistically Receiver Private iOT. We uplift the security of eOT to construct iOT such that it obtains indistinguishability based security against a malicious receiver. If the receiver's choice bit is γ then $\mathsf{m}_{1-\gamma}$ is computationally indistinguishable from a random string to a malicious receiver. We perform this in a round-preserving way by applying the elementary OT (via search OT) to indistinguishability-based security OT transformations from [DGH+20] based on Goldreich-Levin hash function. This yields iOT from the same set of assumptions as eOT.

Three Round Delayed-input Statistically Sender Private $\mathsf{SSP}_{\mathsf{OT}}$. Next, we introduce the notion of delayed-input statistically sender private $\mathsf{SSP}_{\mathsf{OT}}$, where only the last OT message depends on the receiver's choice bit γ. It is a three round OT protocol where the receiver sends the first message as a receiver preprocessing phase, the sender sends the second message as a sender preprocessing phase, and the receiver sends the third message based on γ. The sender obtains random output strings. We carefully apply OT reversal techniques on three-round SRP iOT in a round-preserving way to obtain a version of $\mathsf{SSP}_{\mathsf{OT}}$ where the sender obtains random output bits. Then we combine multiple such bit $\mathsf{SSP}_{\mathsf{OT}}$ protocol with a randomness extractor to obtain the final $\mathsf{SSP}_{\mathsf{OT}}$ protocol. This yields $\mathsf{SSP}_{\mathsf{OT}}$ from the same set of assumptions as iOT (and eOT).

Statistically Sender Private CDS with Preprocessing. We introduce the notion of conditional disclosure of secrets CDS in the preprocessing phase. The first two rounds of CDS are input-independent. The receiver sends the third message which depends on the statement-witness pair. The sender encrypts the plaintext under the statement and sends the ciphertext as the fourth message.

For our 2PC protocol, we require security against a maliciously corrupted statistical receiver and a computationally bounded semi-honest sender. We construct an NC1 circuit which checks the validity of receiver's witness by relying on the result of [KM20]. Then we proceed to combine our three round delayed-input $\mathsf{SSP}_{\mathsf{OT}}$ protocol with information theoretic garbling scheme [Kol05] for NC1 circuit to construct our CDS, where the first two rounds of CDS are the preprocessing phases of $\mathsf{SSP}_{\mathsf{OT}}$. In the third round of the CDS the receiver inputs the witness bits as the choice bit of the $\mathsf{SSP}_{\mathsf{OT}}$ protocol. Upon obtaining the $\mathsf{SSP}_{\mathsf{OT}}$ third round messages, the semi-honest sender garbles an NC1 circuit outputs the plaintext if verification of the receiver's witness succeeds corresponding to the input statement. The sender sends a mapping between the random outputs of $\mathsf{SSP}_{\mathsf{OT}}$ to the wire labels corresponding to the witness bits. The receiver decrypts the wire labels corresponding to the witness bits and evaluates the garbled circuit to obtain the plaintext. This yields our CDS protocol from $\mathsf{SSP}_{\mathsf{OT}}$ and one way functions, obtaining the CDS protocol from the same set of assumptions as eOT.

Two Round Statistically Hiding Commitment. We show that the first two rounds of eOT is a two round statistically hiding commitment where the verifier (acting as the eOT sender) sends the first message as the setup phase. The committer (acting as the eOT receiver) commits to bit γ using the OT second message. Statistical receiver privacy of eOT ensures that statistical hiding of γ. If a corrupt committer breaks binding of the commitment scheme with two valid decommitments corresponding to bits 0 and 1, then those decommitments can be used to break computational sender privacy of eOT by recovering

both sender messages of eOT. This yields two round statistically hiding commitments from the same set of assumptions as eOT.

3 Preliminaries

We present our notations and discuss the building blocks in this section.

3.1 Notations

We denote by $a \leftarrow D$ a uniform sampling of an element a from a distribution D. The set of elements $\{1, \ldots, n\}$ is represented by $[n]$. We denote the computational security parameter by κ and statistical security parameter by μ respectively. Let $\mathbb{Z}_q$ denote the field of order q, where $q = \frac{p-1}{2}$ and p are primes. Let G be the multiplicative group corresponding to $\mathbb{Z}_p^*$ with generator g, where CDH assumption holds. We denote a field of size $\mathcal{O}(2^\mu)$ as $\mathbb{F}$. For a bit $b \in \{0,1\}$, we denote $1-b$ by $\bar{b}$. In our paper we consider one-sided statistical 2PC protocol against explainable parties and also against malicious corruption of parties. We refer to the paper of [KM20] for the one-sided statistical security model against explainable parties and against malicious adversaries for the sake of completeness.

3.2 Oblivious Transfer Protocols

We define our OT notions - eOT, iOT and $\mathsf{SSP}_{\mathsf{OT}}$, as follows.

Elementary OT with Statistical Receiver Privacy (eOT). We denote a three round OT protocol, where sender sends the first message and the sender receives random outputs, by a tuple of four algorithms defined as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$: The sender computes ot_1 as the OT sender message and sends it to the receiver.
- $\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$: The receiver computes the OT receiver message ot_2 and internal state st_R based on choice bit γ and ot_1. The receiver sends ot_2 to the sender.
- $\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2)$: The sender computes $(\mathsf{ot}_3, \mathsf{m}_0, \mathsf{m}_1)$. The sender sends ot_3 as the OT sender message and outputs $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\}$.
- $\mathsf{OT}_\mathsf{R}(\mathsf{st}_\mathsf{R}, \mathsf{ot}_3)$: The receiver computes m' and outputs it.

Correctness. The above three-round OT protocol is said to be correct if for any security parameter $\kappa \in \mathbb{N}$ and any bit $\gamma \in \{0,1\}$, letting

$$\mathsf{ot}_1 \leftarrow \mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa) \quad , \quad (\mathsf{ot}_2, \mathsf{st}_\mathsf{R}) \leftarrow \mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1),$$
$$(\mathsf{ot}_3, \mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2) \quad , \quad \mathsf{m}' \leftarrow \mathsf{OT}_\mathsf{R}(\mathsf{st}_\mathsf{R}, \mathsf{ot}_3),$$

we have $\mathsf{m}' = \mathsf{m}_\gamma$ with overwhelming probability.

Statistical Receiver Privacy. The above OT protocol satisfies statistical receiver privacy if the two tuples are statistically close.

$$\{\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, 0, \mathsf{ot}_1), \mathsf{ot}_1\} \stackrel{s}{\approx} \{\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, 1, \mathsf{ot}_1), \mathsf{ot}_1\},$$

where $\mathsf{ot}_1 \leftarrow \mathcal{A}(1^\kappa)$ is generated by an adversary $\mathcal{A}$ who maliciously corrupts the sender.

Elementary Sender Security. The work of [DGH+20] introduced the notion of elementary sender security in the crs model. It is the weakest security notion against a malicious receiver. We extend their notion to the plain model. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ denote a non-uniform adversary who maliciously corrupts the receiver. To break elementary security the adversary is required to output both strings m_0 and m_1. This is formalized by the following experiment.

$\underline{\mathsf{Exp}^\kappa_{\mathsf{eOT}}(\mathcal{A}) :}$

1. Run $\mathsf{ot}_1 \leftarrow \mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$.
2. Obtain $(\mathsf{ot}_2, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{ot}_1)$.
3. Run $(\mathsf{ot}_3, \mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2)$.
4. Obtain $(\mathsf{m}^*_0, \mathsf{m}^*_1) \leftarrow \mathcal{A}_2(\mathsf{st}_{\mathcal{A}}, \mathsf{ot}_3)$ and output 1 iff $(\mathsf{m}^*_0, \mathsf{m}^*_1) == (\mathsf{m}_0, \mathsf{m}_1)$.

We say that the OT protocol satisfies elementary sender security if $\Pr[\mathsf{Exp}^\kappa_{\mathsf{eOT}}(\mathcal{A}) = 1] = \mathsf{neg}(\kappa)$.

Definition 5 *We denote a three-round OT protocol with the above algorithms as eOT if it satisfies sender elementary security and statistical receiver privacy.*

Indistinguishability OT with Statistical Receiver Privacy (iOT). We denote a three round OT protocol, where sender sends the first message and the parties have chosen inputs, by a tuple of four algorithms defined as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$: The sender computes ot_1 as the OT sender message and sends it to the receiver.
- $\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$: The receiver computes the OT receiver message ot_2 and internal state st_{R} based on choice bit γ and ot_1. The receiver sends ot_2 to the sender.
- $\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{m}_0, \mathsf{m}_1), \mathsf{ot}_2)$: The sender computes ot_3 based on ot_2 and its inputs $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\}$. The sender sends ot_3 as the OT sender message.
- $\mathsf{OT}_{\mathsf{R}}(\mathsf{st}_{\mathsf{R}}, \mathsf{ot}_3)$: The receiver computes m' and outputs it.

Correctness. The above three-round OT protocol is said to be correct if for any security parameter $\kappa \in \mathbb{N}$ and any bit $\gamma \in \{0,1\}$, letting

$$\mathsf{ot}_1 \leftarrow \mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa) \quad , \quad (\mathsf{ot}_2, \mathsf{st}_\mathsf{R}) \leftarrow \mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1),$$
$$\mathsf{ot}_3 \leftarrow \mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{m}_0, \mathsf{m}_1), \mathsf{ot}_2) \quad , \quad \mathsf{m}' \leftarrow \mathsf{OT}_\mathsf{R}(\mathsf{st}_\mathsf{R}, \mathsf{ot}_3),$$

we have $\mathsf{m}' = \mathsf{m}_\gamma$ with overwhelming probability.

Statistical Receiver Privacy. The above OT protocol satisfies statistical receiver privacy if the two tuples are statistically close.

$$\{\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, 0, \mathsf{ot}_1), \mathsf{ot}_1\} \stackrel{s}{\approx} \{\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, 1, \mathsf{ot}_1), \mathsf{ot}_1\},$$

where $\mathsf{ot}_1 \leftarrow \mathcal{A}(1^\kappa)$ is generated by an adversary $\mathcal{A}$ who maliciously corrupts the sender.

Indistinguishability-based Sender Security. Sender's indistinguishability security was defined in [DGH+20] in the crs model. We extend it to the plain model via an experiment $\mathsf{Exp}^{\mathsf{crs},r,w,b}_{\mathsf{iOT}}(\mathcal{A})$ between a non-uniform PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and a challenger, where the experiment is parameterized by random coins $r \in \{0,1\}^\kappa$, a bit $w \in \{0,1\}$, and a bit $b \in \{0,1\}$:

$\underline{\mathsf{Exp}^{w,b}_{\mathsf{iOT}}(\mathcal{A})}$:

1. Run $\mathsf{ot}_1 \leftarrow \mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$.
2. Run $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{ot}_2, \mathsf{st}_\mathcal{A}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{ot}_1; r)$.
3. If $b = 0$, compute $\mathsf{ot}_3 \leftarrow \mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{m}_0, \mathsf{m}_1), \mathsf{ot}_2)$.
4. If $b = 1$, compute $\mathsf{ot}_3 \leftarrow \mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{m}'_0, \mathsf{m}'_1), \mathsf{ot}_2)$ where $m'_w \leftarrow \{0,1\}$ and $m'_{1-w} := m_{1-w}$.
5. Output $s \leftarrow \mathcal{A}_2(\mathsf{st}_\mathcal{A}, \mathsf{ot}_3)$.

We say that iOT satisfies sender's indistinguishability security if for any PPT adversary $\mathcal{A}$, the following holds where the probability is taken over $r \leftarrow \{0,1\}^\kappa$.

$$|\Pr[\mathsf{Exp}^{\mathsf{crs},r,w,0}_{\mathsf{iOT}}(\mathcal{A}) = 1] - \Pr[\mathsf{Exp}^{\mathsf{crs},r,w,1}_{\mathsf{iOT}}(\mathcal{A}) = 1] \leq \mathsf{neg}(\kappa).$$

Definition 6 *We denote a three-round OT protocol with the above algorithms as iOT if it satisfies indistinguishability-based sender security and statistical receiver privacy.*

Statistically Sender Private Random OT ($\mathsf{SSP}_{\mathsf{OT}}$). We denote a three-round OT protocol, where the receiver sends the first message and the sender obtains random outputs, by a tuple of four algorithms defined as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa)$: The receiver computes ot_1 as the OT receiver message and st_R as the internal state. The receiver sends ot_1 to the sender and stores st_R as the internal receiver state.
- $\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_1)$: Given the OT message ot_1, the sender outputs a message ot_2 and secret internal state st_S.
- $\mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}(\mathsf{st}_\mathsf{R}, \gamma, \mathsf{ot}_2)$: Given a secret state st_R, choice bit γ and a message ot_2, the receiver computes $\mathsf{m}' \in \{0,1\}^\ell$ and the OT message ot_3. The receiver sends ot_3 to the sender and outputs m'.
- $\mathsf{OT}_\mathsf{S}(\mathsf{st}_\mathsf{S}, \mathsf{ot}_3)$: Given the secret state st_S and a message ot_3, it outputs two string-messages $(\mathsf{m}_0, \mathsf{m}_1) \in \{0,1\}^\ell$.

Remark. Note that the receiver's choice bit γ is not included in the first algorithm $\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}$ and is only used in the algorithm $\mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}$ thereby allowing the protocol to enjoy a "delayed-input" feature.

Correctness. The above protocol is said to be correct if for any $\kappa \in \mathbb{N}$ and any bit $\gamma \in \{0,1\}$, letting

$$\begin{aligned} (\mathsf{ot}_1, \mathsf{st}_\mathsf{R}) &\leftarrow \mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa) \quad , \qquad (\mathsf{ot}_2, \mathsf{st}_\mathsf{S}) \leftarrow \mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_1), \\ (\mathsf{ot}_3, \mathsf{m}') &\leftarrow \mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}(\mathsf{st}_\mathsf{R}, \gamma, \mathsf{ot}_2) \quad , \qquad (\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathsf{OT}_\mathsf{S}(\mathsf{st}_\mathsf{S}, \mathsf{ot}_3), \end{aligned}$$

we have $\mathsf{m}' = \mathsf{m}_\gamma$ with overwhelming probability.

Computational Receiver Privacy. The above protocol satisfies computational receiver privacy if for any $\kappa \in \mathbb{N}$, any $b \in \{0,1\}$, and any non-uniform PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, letting $\beta = \mathsf{Exp}^{\kappa,b}(\mathcal{A})$, we have

$$|\Pr[\beta = 0] - \Pr[\beta = 1]| \leq \mathsf{negl}(\kappa),$$

where the experiment $\mathsf{Exp}^{\kappa,b}(\mathcal{A})$ is defined as follows:

$\underline{\mathsf{Exp}^{\kappa,b}(\mathcal{A})}$:

1. $(\mathsf{ot}_1, \mathsf{st}_\mathsf{R}) \leftarrow \mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa)$.
2. $(\mathsf{ot}_2, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{ot}_1)$.
3. $(\mathsf{ot}_3, \mathsf{m}') \leftarrow \mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}(\mathsf{st}_\mathsf{R}, b, \mathsf{ot}_2)$.
4. $b' \leftarrow \mathcal{A}_2(\mathsf{ot}_3, \mathsf{st})$.
5. If $b = b'$, output 0. Else, output 1.

Statistical Sender Privacy. Consider an execution of the above three round protocol involving an honest sender and an (unbounded, non-uniform) malicious adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:

$$(\mathsf{ot}_1, \mathsf{st}_\mathsf{R}) \leftarrow \mathcal{A}_1(1^\kappa) \quad , \quad (\mathsf{ot}_2, \mathsf{st}_\mathsf{S}) \leftarrow \mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_1),$$
$$(\mathsf{ot}_3, \mathsf{st}) \leftarrow \mathcal{A}_2(\mathsf{st}_\mathsf{R}, \gamma, \mathsf{ot}_2) \quad , \quad (\mathsf{m}_0, \mathsf{m}_1) \leftarrow \mathsf{OT}_\mathsf{S}(\mathsf{st}_\mathsf{S}, \mathsf{ot}_3).$$

Let $\mathsf{View}^\kappa(\mathcal{A})$ denote the view of the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the above protocol execution. A three-round SSP-string-sROT protocol is said to satisfy statistical sender privacy if for any $\kappa \in \mathbb{N}$ and any (unbounded, non-uniform) adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a bit $\beta \in \{0, 1\}$ such that the following two distributions are statistically indistinguishable:

$$(\mathsf{View}^\kappa(\mathcal{A}), \mathsf{m}_\beta) \stackrel{s}{\approx} (\mathsf{View}^\kappa(\mathcal{A}), \mathsf{U}),$$

where $\mathsf{U} \leftarrow \{0,1\}^{|\mathsf{m}_\beta|}$ denotes a random bit string of size $|\mathsf{m}_\beta|$.

Definition 7 *We denote a three-round OT protocol with the above algorithms as $\mathsf{SSP}_{\mathsf{OT}}$ if it satisfies statistical sender privacy and computational receiver privacy.*

3.3 Garbling Schemes

A garbling scheme [Yao86, LP09, BHR12] consists of the following algorithms: Gb takes a circuit $\mathcal{C}$ as input and outputs a garbled circuit GC, encoding information Keys, and decoding information d. En takes an input x and encoding information Keys and outputs a garbled input X. Ev takes a garbled circuit and garbled input X and outputs a garbled output Y. Finally, De takes a garbled output Y and decoding information and outputs a plain circuit-output (or an error, $\perp$). There is an additional verification algorithm Ve in the garbling scheme which when accepts a given $(\mathrm{GC}, \mathsf{Keys}, \mathsf{d})$ signifies that the GC is correct, and that the garbled output corresponding to any clear output can be extracted. Formally, a *garbling scheme* is defined by a tuple of functions $\mathsf{Garble} = (\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \mathsf{De}, \mathsf{Ve})$, described as follows:

- $\mathsf{Gb}\,(1^\kappa, \mathcal{C}) = (\mathrm{GC}, \mathsf{Keys}, \mathsf{d})$: A randomized algorithm which takes as input the security parameter and a circuit $\mathcal{C} : \{0,1\}^n \to \{0,1\}^m$ and outputs a tuple of strings $(\mathrm{GC}, \mathsf{Keys}, \mathsf{d})$, where GC is the garbled circuit, Keys denotes the input-wire labels, and d denotes the decoding information.
- $\mathsf{En}\,(x, \mathsf{Keys}) = \mathsf{X}$: a deterministic algorithm that outputs the garbled input X corresponding to input x.
- $\mathsf{Ev}\,(\mathrm{GC}, \mathsf{X}) = \mathsf{Y}$: A deterministic algorithm which evaluates garbled circuit GC on garbled input X, and outputs a garbled output Y.
- $\mathsf{De}\,(\mathsf{Y}, \mathsf{d}) = y$: A deterministic algorithm that outputs the plaintext output corresponding to Y or $\perp$ signifying an error if the garbled output Y is invalid.
- $\mathsf{Ve}\,(\mathcal{C}, \mathrm{GC}, \mathsf{Keys}, \mathsf{d}) = 1 or \perp$: A deterministic algorithm which takes as input a circuit $\mathcal{C} : \{0,1\}^n \mapsto \{0,1\}^m$, a garbled circuit (possibly malicious) GC, encoding information e and decoding information d. It outputs 1 when GC is a valid garbling of $\mathcal{C}$, and $\perp$ otherwise.

The garbling scheme used in our protocols need to satisfy several properties such as *correctness, privacy, verifiability and reconstructability.* We borrow the notations from the work of [HV16].

Definition 8 (Perfect Correctness). *A garbling scheme* Garble *is perfectly correct if for all input lengths* $n \leq \mathsf{poly}(\kappa)$*, circuits* $\mathcal{C} : \{0,1\}^n \to \{0,1\}^m$ *and inputs* $x \in \{0,1\}^n$*, the following holds:*

$$\Pr\left[\mathsf{De}(\mathsf{Ev}(\mathrm{GC}, \mathsf{En}(\mathit{Keys}, x)), \mathit{d}) \neq \mathcal{C}(x) : (\mathrm{GC}, \mathit{Keys}, \mathit{d}) \leftarrow \mathsf{Gb}(1^\kappa, \mathcal{C})\right] = 1.$$

Definition 9 (Privacy). *A garbling scheme* Garble *is private if for all input lengths* $n \leq \mathsf{poly}(\kappa)$*, circuits* $\mathcal{C} : \{0,1\}^n \to \{0,1\}^m$*, there exists a PPT simulator* Sim *such that for all inputs* $x \in \{0,1\}^n$*, for all adversaries* $\mathcal{A}$*, the following two distributions are statistically indistinguishable:*

- $\text{REAL}(\mathcal{C}, x)$ *: run* $(\mathrm{GC}, \mathit{Keys}, \mathit{d}) \leftarrow \mathsf{Gb}(1^\kappa, \mathcal{C})$*, and output* $(\mathrm{GC}, \mathsf{En}(x, \mathit{Keys}), \mathit{d})$*.*
- $\text{IDEAL}_{\mathsf{Sim}}(\mathcal{C}, \mathcal{C}(x))$*: Compute* $(\mathrm{GC}', \mathsf{X}, \mathit{d}', \mathit{st}) \leftarrow \mathcal{S}_{GC}(1^\kappa, \mathcal{C}, \mathcal{C}(x))$ *and output* $(\mathrm{GC}', \mathsf{X}, \mathit{d}')$*.*

Definition 10 (Verifiability). *A garbling scheme* Garble *is verifiable if for all input lengths* $n \leq \mathit{poly}(\kappa)$*, circuits* $\mathcal{C} : \{0,1\}^n \to \{0,1\}^m$*, inputs* $x \in \{0,1\}^n$*, and adversary* $\mathcal{A}$*, the following probability is negligible in* κ*:*

$$\Pr\left(\mathsf{De}(\mathsf{Ev}(\mathrm{GC}, \mathsf{En}(x, \mathit{Keys})), \mathit{d}) \neq \mathcal{C}(x) \ : \ \begin{array}{c}(\mathrm{GC}, \mathit{Keys}, \mathit{d}) \leftarrow \mathcal{A}(1^\kappa, \mathcal{C}) \\ \mathsf{Ve}\left(\mathcal{C}, \mathrm{GC}, \mathit{Keys}, \mathit{d}\right) = 1 \neq \bot\end{array}\right).$$

We are interested in a class of garbling schemes referred to as *projective* in [BHR12]. When garbling a circuit $\mathcal{C} : \{0,1\}^n \mapsto \{0,1\}^m$, a projective garbling scheme produces encoding information of the form $\mathsf{Keys} = \left(\mathsf{Keys}_i^0, \mathsf{Keys}_i^1\right)_{i\in[n]}$, and the encoded input X corresponding to $x = (x_i)_{i\in[n]}$ can be interpreted as $\mathsf{X} = \mathsf{En}(x, \mathsf{Keys}) = (\mathsf{Keys}_i^{x_i})_{i\in[n]}$. Information-theoretic Garbled circuits for NC1 circuits with information theoretic privacy can be built from one way functions [Yao86, LP09] based on one-way functions satisfies.

3.4 Zero-Knowledge Proofs and Arguments for NP [KM20]

An n-round delayed-input interactive protocol for deciding a language L corresponding to a relation $\mathcal{R}$ is denoted by $\langle \mathsf{P}, \mathsf{V}\rangle$ and it proceeds as follows:

- At the beginning of the protocol, P and V receive the size of the instance and execute the first $n-1$ rounds.
- At the start of the last round, P receives input $(x, w) \in \mathcal{R}$ and V receives x. Upon receiving the last round message from P, V outputs 0 or 1.

For our protocols, we rely on proofs and arguments for NP that satisfy delayed-input completeness, adaptive soundness and adaptive ZK. Fix any language L. Let $\langle \mathsf{P}, \mathsf{V}\rangle$ denote the execution of a protocol between a PPT prover P and a (possibly unbounded) verifier V, let $\mathsf{V}_{\mathsf{out}}$ denote the output of the verifier and let $\mathsf{View}_{\mathcal{A}}\langle \mathsf{P}, \mathsf{V}\rangle$ denote the transcript together with the state and randomness of a party $\mathcal{A} \in \{\mathsf{P}, \mathsf{V}\}$ at the end of an execution of a protocol.

Definition 11 (Statistical Zero Knowledge Argument). *For any fixed language L, we say $\langle \mathsf{P}, \mathsf{V} \rangle$ is a delayed-input statistical zero knowledge argument system for L if the following properties hold:*

- *Completeness: For all $x \in \mathsf{L}$,*

$$\Pr[\mathsf{V}_{out}\langle \mathsf{P}, \mathsf{V} \rangle = 1] = 1 - \mathsf{neg}(\kappa),$$

  where the probability is over the random coins of P and V.

- *Adaptive Soundness: For all polynomial size P^* and all $x \notin \mathsf{L}$ sampled by P^* adaptively depending upon the first $n-1$ rounds,*

$$\Pr[\mathsf{V}_{out}\langle \mathsf{P}^*, \mathsf{V} \rangle = 1] = \mathsf{neg}(\kappa).$$

- *Statistical Zero Knowledge: There exists a PPT simulator Sim such that for all V^* and all $x \in \mathsf{L}$,*

$$\left| \Pr[\mathsf{V}^*(\mathsf{View}_{\mathcal{A}}\langle \mathsf{P}(x,w), \mathsf{V}^* \rangle) = 1] - \Pr[\mathsf{V}^*(\mathsf{Sim}^{\mathsf{V}^*}(x)) = 1] \right| = \mathsf{neg}(\kappa).$$

Four round delayed-input statistical zero knowledge arguments can be obtained from [LS91] by relying on two round statistically hiding commitments.

Definition 12 (Zero Knowledge Proof). *For any fixed language L, we say $\langle \mathsf{P}, \mathsf{V} \rangle$ is a delayed-input zero knowledge proof system for L if the following properties hold:*

- *Completeness: For all $x \in \mathsf{L}$,*

$$\Pr[\mathsf{V}_{out}\langle \mathsf{P}, \mathsf{V} \rangle = 1] = 1 - \mathsf{neg}(\kappa),$$

  where the probability is over the random coins of P and V.

- *Adaptive Soundness: For all P^* and all $x \notin \mathsf{L}$ sampled by P^* adaptively depending upon the first $n-1$ rounds,*

$$\Pr[\mathsf{V}_{out}\langle \mathsf{P}^*, \mathsf{V} \rangle = 1] = \mathsf{neg}(\kappa).$$

- *Computational Zero Knowledge: There exists a PPT simulator Sim such that for all polynomial size V^* and all $x \in \mathsf{L}$,*

$$\left| \Pr[\mathsf{V}^*(\mathsf{View}_{\mathcal{A}}\langle \mathsf{P}(x,w), \mathsf{V}^* \rangle) = 1] - \Pr[\mathsf{V}^*(\mathsf{Sim}^{\mathsf{V}^*}(x)) = 1] \right| = \mathsf{neg}(\kappa).$$

Five round delayed-input zero knowledge proofs can be obtained by relying on [GK96], where the instance is adaptively chosen in the last round by the combining techniques from [LS91]. The proof system can be instantiated from two round statistically hiding commitments.

3.5 Low Depth-Proofs

The work [KM20] described how any computation that is verifiable by a family of polynomial sized circuits can be transformed into a proof that is verifiable by a family of circuits in NC1. Let $\mathcal{R}$ be an efficiently computable binary relation. For pairs $(x, w) \in \mathcal{R}$, denote x as the statement and w as the witness. Let L be the language consisting of statements in $\mathcal{R}$.

Definition 13 (Low-Depth Non-Interactive Proofs). *A low-depth non interactive proof with perfect completeness and soundness for a relation $\mathcal{R}$ consists of a PPT prover P and a verifier V that satisfy:*

- *Perfect completeness: A proof system is perfectly complete if an honest prover with a valid witness can always convince an honest verifier. For all $(x, w) \in \mathcal{R}$ we have,*
$$\Pr[\mathsf{V}(\pi) = 1 | \pi \leftarrow \mathsf{P}(x, w)] = 1.$$
- *Perfect soundness: A proof system is perfectly sound if it is infeasible to convince an honest verifier when the statement is false. For all $x \notin \mathsf{L}$ and all (even unbounded) adversaries $\mathcal{A}$ we have*
$$\Pr[\mathsf{V}(x, \pi) = 1 | \pi \leftarrow \mathcal{A}(x)] = 0.$$
- *Low Depth: The verifier algorithm V can be implemented in NC1.*

The works of [GGH+13] and [KM20] presented a simple construction of a low-depth non-interactive proof for any NP-verification circuit. The prover P executes the NP-verification circuit on the witness and generates the proof as the sequential concatenation (in some specified order) of the bit values assigned to the individual wires of the circuit. The verifier V proceeds by checking consistency of the values assigned to the internal wires of the circuit for each gate. In particular for each gate in the NP-verification circuit the verifier checks if the wire vales provided in the proof represent a correct evaluation of the gate. Since the verification corresponding to each gate can be done independent of every other gate and in constant depth, we have that V itself is constant depth.

4 Three Round Oblivious Transfer Protocols

In this section, we describe our statistically sender private $\mathsf{SSP}_{\mathsf{OT}}$ construction from eOT which satisfies statistical receiver privacy. First, we build iOT from eOT and then we build $\mathsf{SSP}_{\mathsf{OT}}$ from iOT. All our protocols are round preserving in nature. The corresponding definitions of the OT protocols can be found in 3.2. Our $\mathsf{SSP}_{\mathsf{OT}}$ protocol enjoys a delayed-input feature since only the last OT protocol message depends on the receiver's input. This will be useful later on in obtaining statistically sender private CDS in the preprocessing model and also our one-sided statistical 2PC.

4.1 Statistically Receiver Private Indistinguishability-based OT

We denote an elementary OT protocol as $\mathsf{eOT} = (\mathsf{eOT}.\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{eOT}.\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{eOT}.\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{eOT}.\mathsf{OT}_{\mathsf{R}})$. We construct our indistinguishability based SRP-bit OT protocol, denoted as iOT, as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$: The sender obtains $\mathsf{eOT.ot}_1 \leftarrow \mathsf{eOT.OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$. The sender sends $\mathsf{ot}_1 = \mathsf{eOT.ot}_1$ as the OT sender message.
- $\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$: The receiver computes the OT receiver message as $(\mathsf{eOT.ot}_2, \mathsf{eOT.st}_\mathsf{R}) \leftarrow \mathsf{eOT.OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$. It sends $\mathsf{ot}_2 = \mathsf{eOT.ot}_2$ to the sender and stores $\mathsf{st}_\mathsf{R} = \mathsf{eOT.st}_\mathsf{R}$.
- $\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{m}_0, \mathsf{m}_1), \mathsf{ot}_2)$: The sender performs the following:
  - The sender runs eOT sender protocol for κ times on ot_2 to compute
    $$\{\mathsf{eOT.ot}_{3,i}, (\mathsf{eOT.m}_{0,i}, \mathsf{eOT.m}_{1,i})\} = \mathsf{eOT.OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2),$$
    for $i \in [\kappa]$.
  - Sender computes $\mathsf{eOT.m}_\alpha = (\mathsf{eOT.m}_{\alpha,1}, \ldots \mathsf{eOT.m}_{\alpha,\kappa})$ for $\alpha \in \{0,1\}$.
  - Denote the length of $\mathsf{eOT.m}_0$ and $\mathsf{eOT.m}_1$ as $n = n(\kappa)$ where $n = |\mathsf{eOT.m}_0| = |\mathsf{eOT.m}_1|$.
  - The sender samples $s_0, s_1 \leftarrow \{0,1\}^n$ as the description of the Goldreich-Levin Hash function.
  - The sender computes $\mathsf{p}_\alpha = \mathsf{m}_\alpha \oplus \langle \mathsf{eOT.m}_\alpha, s_\alpha \rangle$ for $\alpha \in \{0,1\}$.

  The sender sends $\mathsf{ot}_3 = (\{\mathsf{eOT.ot}_{3,i}\}_{i\in[\kappa]}, s_0, s_1, \mathsf{p}_0, \mathsf{p}_1)$.
- $\mathsf{OT}_\mathsf{R}(\mathsf{st}_\mathsf{R}, \mathsf{ot}_3)$: The receiver performs the following:
  - The receiver runs eOT decryption algorithm for κ times to compute
    $$\{\mathsf{eOT.m}_{\gamma,i}\} = \mathsf{eOT.OT}_\mathsf{R}(\mathsf{eOT.st}_\mathsf{R}, \mathsf{eOT.ot}_{3,i}),$$
    for $i \in [\kappa]$.
  - The receiver sets $\mathsf{eOT.m}_\gamma = (\mathsf{eOT.m}_{\gamma,1}, \ldots, \mathsf{eOT.m}_{\gamma,\kappa})$.
  - The receiver outputs $\mathsf{m}_\gamma = \mathsf{p}_\gamma \oplus \langle \mathsf{eOT.m}_\gamma, s_\gamma \rangle$.

Correctness. It can be verified in a straightforward manner.

Lemma 14 *The above protocol satisfies perfect receiver's privacy if eOT satisfies perfect receiver privacy.*

Proof. The receiver's choice bit γ is perfectly hidden in OT message $\mathsf{ot}_2 = \mathsf{eOT.ot}_2$ if $\mathsf{eOT.ot}_2$ perfectly hides γ. □

Lemma 15 *The above protocol satisfies sender's indistinguishability based security if eOT satisfies computational sender's elementary security.*

Proof. The work of [DGH+20] showed that the above transformation converts an elementary OT to an iOT OT protocol (via search OT). By combining Theorems 5.2 and 5.3 of [DGH+20] we prove the above theorem. We refer to their paper for more details regarding the proof steps. □

4.2 Three round Statistically Sender Private OT

Our $\mathsf{SSP}_{\mathsf{OT}}$ construction relies on randomness extractors and the leftover hash lemma. We briefly define them as follows for completeness.

Definition 16 *(Randomness Extractor.)* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ *is a strong* (k, ϵ) *randomness extractor if for every* k*-source* $X \in \{0,1\}^n$ *the following holds:*

$$\{U_d, \mathsf{Ext}(X, U_d)\} \overset{\epsilon}{\approx} \{U_d, U_\ell\},$$

where U_d *and* U_ℓ *are uniformly sampled* d*-bit and* ℓ*-bit strings respectively.*

Definition 17 *(Leftover Hash Lemma.) If* $\mathrm{H} = \{h : \{0,1\}^n \to \{0,1\}^\ell\}$ *is a pairwise independent hash family of hash function where* $\ell = k - 2\log_2(\frac{1}{\epsilon})$*, then* $\mathsf{Ext}(x, h) \overset{def}{=} h(x)$ *is a strong* (k, ϵ) *extractor.*

Construction. We denote an iOT protocol as $\mathsf{iOT} = (\mathsf{iOT.OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{iOT.OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{iOT.OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{iOT.OT}_{\mathsf{R}})$. We define our SSP-OT $\mathsf{SSP}_{\mathsf{OT}}$ as a tuple of four algorithms defined as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa)$:
  - The receiver runs iOT protocol for n times by computing $\{\mathsf{iOT.ot}_{1,i}\} = \mathsf{iOT.OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$ for $i \in [n]$.
  - The receiver sends $\mathsf{ot}_1 = \{\mathsf{iOT.ot}_{1,i}\}_{i\in[n]}$ as the OT receiver message.
- $\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_1)$: The sender performs the following for $i \in [n]$:
  - The sender samples $\gamma_i \leftarrow \{0,1\}$.
  - The sender computes $(\mathsf{iOT.ot}_{2,i}, \mathsf{iOT.st}_{\mathsf{R},i}) = \mathsf{iOT.OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma_i, \mathsf{iOT.ot}_{1,i})$ with choice bit set to γ_i.
  - The sender samples a mapping $\mathsf{Map}_i \leftarrow \{0,1\}$.
  - The sender samples a pairwise independent hash function $h \leftarrow \mathrm{H}_\kappa$.

  The sender sends $\mathsf{ot}_2 = (h, \{\mathsf{iOT.ot}_{2,i}, \mathsf{Map}_i\}_{i\in[n]})$ as the OT sender message and stores $\mathsf{st}_{\mathsf{S}} = \{\mathsf{iOT.st}_{\mathsf{R},i}, \mathsf{Map}_i, \gamma_i\}_{i\in[n]}$ as the internal state.
- $\mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, b, \mathsf{ot}_2)$:
  - The receiver samples $\mathsf{p}_{0,i} \leftarrow \{0,1\}$ and sets $\mathsf{p}_{1,i} = b \oplus \mathsf{p}_{0,i}$ for every $i \in [n]$. The receiver computes $\mathsf{iOT.ot}_{3,i} = \mathsf{iOT.OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{p}_{0,i}, \mathsf{p}_{1,i}), \mathsf{iOT.ot}_{2,i})$ for every $i \in [n]$.
  - The receiver sets $\mathsf{t}_{b,i} = \mathsf{Map}_i \oplus \mathsf{p}_{0,i}$.
  - The receiver sets $\mathsf{t}_b = (\mathsf{t}_{b,1}, \ldots, \mathsf{t}_{b,n})$.
  - The receiver computes $\mathsf{m}_b = H(\mathsf{t}_b)$.

The receiver sends $\mathsf{ot}_3 = \{\mathsf{iOT.ot}_{3,i}\}_{i\in[n]}$ as the OT receiver message and outputs m_b as the output.

- $\mathsf{OT_S}(\mathsf{st_S}, \mathsf{ot}_3)$:
  - The sender computes $\mathsf{a}_i = \mathsf{iOT.OT_R}(\mathsf{iOT.st}_{\mathsf{R},i}, \mathsf{iOT.ot}_{3,i})$ for $i \in [n]$.
  - The sender computes $\mathsf{t}_{i,0} = \mathsf{Map}_i \oplus \mathsf{a}_i$ and $\mathsf{t}_{i,1} = \gamma_i \oplus \mathsf{t}_{i,0}$.
  - For $\alpha \in \{0,1\}$, the sender sets $\mathsf{t}_\alpha = (\mathsf{t}_{\alpha,1}, \ldots, \mathsf{t}_{\alpha,n})$.
  - For $\alpha \in \{0,1\}$, the sender computes $\mathsf{m}_\alpha = H(\mathsf{t}_\alpha)$.

  The sender outputs $(\mathsf{m}_0, \mathsf{m}_1)$.

Correctness. The sender computes p_{i,γ_i} from the ith iOT run. The sender sets $\mathsf{t}_{0,i} = \mathsf{Map}_i \oplus \mathsf{a}_i = \mathsf{Map}_i \oplus \mathsf{p}_{\gamma_i,i}$ and $\mathsf{t}_{1,i} = \gamma_i \oplus \mathsf{t}_{0,i} = \gamma_i \oplus \mathsf{Map}_i \oplus \mathsf{p}_{\gamma_i,i}$. The receiver computes the following from the ith OT:

$$\begin{aligned}
(b', \mathsf{t}'_{b_i}) &= (\mathsf{p}_{0,i} \oplus \mathsf{p}_{1,i}, \mathsf{Map}_i \oplus \mathsf{p}_{0,i}) \\
&= (\mathsf{p}_{0,i} \oplus \mathsf{p}_{1,i}, \mathsf{Map}_i \oplus \mathsf{p}_{\gamma_i,i} \oplus (\mathsf{p}_{0,i} \oplus \mathsf{p}_{1,i}) \cdot \gamma_i) \\
&= (b, \mathsf{Map}_i \oplus \mathsf{p}_{\gamma_i,i} \oplus b \cdot \gamma_i) \\
&= (b, \mathsf{t}_{0,i} \oplus b \cdot \gamma_i) \\
&= (b, \mathsf{t}_{b,i}).
\end{aligned}$$

The sender outputs $\mathsf{m}_\alpha = H(\mathsf{t}_\alpha)$ for $\alpha \in \{0,1\}$. And the receiver outputs $\mathsf{m}_b = H(\mathsf{t}_b)$ thus proving correctness.

Lemma 18 *The above protocol satisfies statistical sender privacy if iOT satisfies statistical receiver privacy and H is a $(\lceil \frac{n}{2} \rceil, \epsilon)$-randomness extractor.*

Proof. The sender's secret input γ_i to the ith iOT remains hidden due to statistical receiver privacy of iOT. Without loss of generality, assuming a corrupt receiver obtains atmost $\lceil \frac{n}{2} \rceil$ bits of t_0 and $\lfloor \frac{n}{2} \rfloor$ bits of t_1 simultaneously by setting $b == 0$ for $\frac{n}{2}$ runs of iOT and setting $b == 1$ for the rest $\frac{n}{2}$ runs of iOT. In such a case, $\lceil \frac{n}{2} \rceil$ bits of t_1 remains hidden and is uniformly distributed. Thus the input space of the hash function H has an entropy of $k = \lceil \frac{n}{2} \rceil$ and $\ell = \lceil \frac{n}{2} \rceil - 2\log_2(\frac{1}{\epsilon})$. Applying the leftover hash lemma we argue that H behaves as a (k, ϵ) randomness extractor and thus statistically hiding m_1. The same argument holds for statistically hiding m_0 if the receiver sets $b == 1$ in $\lceil \frac{n}{2} \rceil$ runs of iOT. □

Lemma 19 *The above protocol satisfies computational receiver privacy if iOT satisfies computational sender privacy.*

Proof. We demonstrate that execution of the protocol with choice bit $b == 0$ is indistinguishable from the execution of the protocol with choice bit $b == 1$ through a sequence of $n + 1$ hybrids $\{\mathtt{Hyb}_j\}_{j\in[n+1]}$. $\mathtt{Hyb}_1$ corresponds to execution of the protocol with choice bit $b == 0$. $\mathtt{Hyb}_{n+1}$ corresponds to execution of the protocol with choice bit $b == 1$. We define the jth hybrid $\mathtt{Hyb}_i$ as follows:

Hybrid $\mathtt{Hyb}_j$:

- $\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa)$:
  - The receiver runs iOT protocol for n times by computing $\{\mathsf{iOT.ot}_{1,i}\} = \mathsf{iOT.OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$ for $i \in [n]$.
  - The receiver sends $\mathsf{ot}_1 = \{\mathsf{iOT.ot}_{1,i}\}_{i\in[n]}$ as the OT receiver message.
- $\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_1)$: The sender sends $\mathsf{ot}_2 = (h, \{\mathsf{iOT.ot}_{2,i}, \mathsf{Map}_i\}_{i\in[n]})$ as the OT sender message and stores $\mathsf{st}_\mathsf{S} = \{\mathsf{iOT.st}_{\mathsf{R},i}, \mathsf{Map}_i, \gamma_i\}_{i\in[n]}$ as the internal state.
- $\mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \mathsf{ot}_2)$:
  - The receiver computes $\mathsf{iOT.ot}_{3,i} = \mathsf{iOT.OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{p}_{0,i}, \mathsf{p}_{1,i}), \mathsf{iOT.ot}_{2,i})$ where $\mathsf{p}_{0,i} \leftarrow \{0,1\}$ and $\mathsf{p}_{1,i} = \mathsf{p}_{0,i}$ for $i \in [j]$,
  - The receiver computes $\mathsf{iOT.ot}_{3,i} = \mathsf{iOT.OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (\mathsf{p}_{0,i}, \mathsf{p}_{1,i}), \mathsf{iOT.ot}_{2,i})$ where $\mathsf{p}_{0,i} \leftarrow \{0,1\}$ and $\mathsf{p}_{1,i} = \mathsf{p}_{0,i} \oplus 1$ for $i \in [j+1, n]$.

  The receiver sends $\mathsf{ot}_3 = \{\mathsf{iOT.ot}_{3,i}\}_{i\in[n]}$ as the OT receiver message and outputs m_b as the output.
- $\mathsf{OT}_\mathsf{S}(\mathsf{st}_\mathsf{S}, \mathsf{ot}_3)$: The sender performs its own adversarial algorithm.

A corrupt sender distinguishing between $\mathtt{Hyb}_j$ and $\mathtt{Hyb}_{j+1}$ breaks computational sender privacy of the jth iOT. □

5 One-Sided Statistically Secure 2PC against Explainable Parties

We describe our one-sided statistically secure 2PC protocol π_{exp} secure against explainable parties in this section. High level overview can be found in Sec. 2.1.

5.1 Protocol π_{exp}

The work of [KM20] built a 2PC protocol against explainable parties. We recall their result as follows.

Theorem 20 *[KM20] Assuming the following holds:*

1. *Three round statistically binding and computationally hiding commitment scheme satisfying extractability,*
2. *Four round statistically hiding and computationally binding commitment scheme satisfying extractability,*
3. *Information theoretic garbled circuits for NC1 circuits,*
4. *Three round oblivious transfer with statistical privacy for a receiver and computational privacy for a sender,*

there exists a robust 5-round secure two-party computation protocol π_{exp} with black-box simulation against unbounded explainable receivers and PPT explainable senders, where the receiver obtains its output at the end of fourth round and the sender obtains its output at the end of the fifth round.

We demonstrate that our elementary OT protocol (with statistical receiver privacy) and a non-interactive commitment/public key encryption scheme with perfect decryption suffices to instantiate the above primitives:

1. Three round statistically binding and computationally hiding commitments can be based on any non-interactive commitment scheme [PRS02], which can itself be based on any public-key encryption [LS19] (satisfying perfect correctness) or injective one-way function [Blu81].
2. Four round statistically hiding and computationally binding commitment scheme satisfying extractability can be obtained from two round statistically hiding commitment schemes which we build from iOT in Sec. 5.2.
3. Garbled circuits can be obtained [Yao86] from one way functions.
4. The three round SRP-OT protocol is instantiated using the iOT protocol from Sec. 4.1.

Non-interactive commitments can be built [Blu81] from injective one-way functions. Injective OWFs can be constructed from Discrete Log and hence CDH[7]. Injective OWFs can also be constructed from computational CSIDH assumption [CLM+18] and hence reciprocal and decisional CSIDH assumptions since the former two assumptions imply [CLM+18, ADMP20, LGdSG21] computational CSIDH. Moreover, it can be shown that the first OT message from the receiver in a 2 round statistically sender private OT protocol acts as a non-interactive commitment, hence yielding additional instantiations from LWE, QR, N^{th} residuosity, DDH, decisional CSIDH and LPN+derandomization. Next, we build two round statistically hiding commitments from iOT. iOT has been constructed from eOT in Sec. 4.1.

5.2 Two round Statistically Hiding Commitment

We denote an iOT protocol as $\mathsf{iOT} = (\mathsf{iOT.OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{iOT.OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{iOT.OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{iOT.OT}_{\mathsf{R}})$. We define a two round statistically hiding commitment Com as tuple of three algorithms $(\mathsf{Com}_1, \mathsf{Com}_2, \mathsf{Decom})$ between a sender and a receiver as follows:

- $\mathsf{Com}_1(1^\kappa)$: The receiver computes $c_1 = \mathsf{iOT.ot}_1 = \mathsf{iOT.OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$. The receiver sends c_1 as the first message of the commitment scheme.
- $\mathsf{Com}_2(1^\kappa, c_1, b)$: The sender computes $(c_2, d) = \mathsf{iOT.OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, b, c_1)$. The sender sends c_2 as the commitment and stores $\mathsf{st} = (b, d)$ as the decommitment.

[7]The adversary is required to output a given tuple (g, g^a) where $a \in \mathbb{Z}_q$ and $g \in G$ are randomly sampled by the challenger

- $\mathsf{Decom}(\mathsf{st}, (c_1, c_2))$: The sender sends $\mathsf{st} = (b, d)$ as the decommitment. The receiver performs the following for $i \in [\kappa]$:
  - Computes $(\mathsf{iOT.ot}_3^i, (\mathsf{m}_0^i, \mathsf{m}_1^i)) = \mathsf{iOT.OT}_{\mathsf{S}\to\mathsf{R}}^{(3)}(1^\kappa, c_1)$.
  - The receiver aborts if $\mathsf{iOT.OT}_{\mathsf{R}}(\mathsf{st}, \mathsf{iOT.ot}_3^i) \neq \mathsf{m}_b^i$.

  The receiver outputs accept if the above checks pass.

Theorem 21 *$Com = (Com_1, Com_2, Decom)$ is a two round statistically hiding commitment scheme with computational binding if iOT satisfies statistical receiver privacy and computational sender security.*

Proof. We argue hiding and binding of Com as follows:

- The sender's committed bit b remains statistically hidden in c_2 since c_2 is the output of $\mathsf{iOT.OT}_{\mathsf{R}\to\mathsf{S}}^{(2)}$ algorithm and c_2 statistically hides b due to statistical receiver privacy of iOT.
- If a corrupt receiver breaks binding of the protocol by producing two valid openings $(0, d_0)$ and $(1, d_1)$ then it breaks sender privacy of the iOT protocol. m_0^i (resp. m_1^i) can be correctly decrypted using $(0, d_0)$ (resp. $(1, d_1)$) as the receiver's decryption randomness.

□

6 One-Sided Statistically Secure 2PC against Malicious Corruptions

We describe our one-sided statistically secure 2PC protocol π_{mal} secure against malicious corruption of parties in this section. We rely on the following primitives for our protocol.

1. Five round one-sided statistically secure 2PC protocol against explainable parties where both parties get the output. We instantiate it using π_{exp} (Thm. 20) based on eOT and non-interactive commitments.
2. Four round statistically sender private Conditional Disclosure of Secrets, denoted as CDS, in the preprocessing phase where the first two rounds are input-independent.
3. Four round delayed-input statistical zero knowledge SZK. This can be built [LS91] from two round statistically hiding commitment.
4. The five round delayed-input zero knowledge proof ZKP. This can be obtained [LS91]+ [GK96] from two round statistically hiding commitment.
5. Four round statistically sender private Conditional Disclosure of Secrets, denoted as CDS, in the preprocessing phase where the first two rounds are input-independent.

The two round statistically hiding commitment is built from eOT (via iOT) in Sec. 5.2. Next, we formally define and construct the CDS protocol before proceeding to the construction of π_{mal}.

6.1 Conditional Disclosure of Secrets in the Preprocessing Model

We denote a Conditional Disclosure of Secrets in preprocessing model as a tuple of five algorithms $\mathsf{CDS} = (\mathsf{CDS}_1, \mathsf{CDS}_2, \mathsf{CDS}_3, \mathsf{CDS}_4, \mathsf{CDS}_5)$ defined as follows:

- $\mathsf{CDS}_1(1^\kappa)$: The receiver computes $(\mathsf{cds}_1, \mathsf{st}_\mathsf{R})$ in the preprocessing phase. The receiver sends cds_1 and stores st_R as the internal state.
- $\mathsf{CDS}_2(1^\kappa, \mathsf{cds}_1)$: The sender computes $(\mathsf{cds}_2, \mathsf{st}_\mathsf{S})$ in the preprocessing phase. The sender sends cds_2 and stores st_S as internal state.
- $\mathsf{CDS}_3(1^\kappa, (x, w), \mathsf{st}_\mathsf{R}, \mathsf{cds}_2)$: The receiver computes $(\mathsf{cds}_3, \mathsf{st}_\mathsf{R})$ based on the statement x, witness w and cds_2. The receiver sends cds_3 and updates st_R as the internal state.
- $\mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3)$: The sender encrypts plaintext ptxt based on statement x and cds_3 to compute cds_4. The sender sends cds_4.
- $\mathsf{CDS}_5(\mathsf{st}_\mathsf{R}, \mathsf{cds}_4)$: The receiver outputs ptxt' as the decrypted message.

The above algorithms should satisfy the following properties:

Correctness. For any $(x, w) \in \mathcal{L}$, and message $\mathsf{ptxt} \in \{0,1\}^*$ the following holds:

$$\Pr\Big[\mathsf{CDS}.\mathsf{CDS}_5(\mathsf{st}_\mathsf{R}, \mathsf{cds}_4) == \mathsf{ptxt} \Big| (\mathsf{cds}_1, \mathsf{st}_\mathsf{R}) \leftarrow \mathsf{CDS}_1(1^\kappa), (\mathsf{cds}_2, \mathsf{st}_\mathsf{S}) \leftarrow \mathsf{CDS}_2(1^\kappa, \mathsf{cds}_1),$$

$$(\mathsf{cds}_3, \mathsf{st}_\mathsf{R}) \leftarrow \mathsf{CDS}_3(1^\kappa, (x, w), \mathsf{st}_\mathsf{R}, \mathsf{cds}_2), \mathsf{cds}_4 \leftarrow \mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3)\Big] == 1$$

Message Indistinguishability. For any $x \notin \mathcal{L}$, $\mathsf{cds}_3^* \in \{0,1\}^*$ and any two equal-length messages ptxt_0, ptxt_1, the following distributions are statistically indistinguishable:

$$\mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}_0), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3^*) \overset{s}{\approx} \mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}_1), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3^*)$$

Receiver Simulation. There exists a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that for any PPT distinguisher $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$, such that for any $x \in \mathcal{L}$, with $\mathcal{R}(x, w) = 1$ the following holds:

$$\Big|\Pr[\mathcal{D}_2(\mathsf{CDS}_3(1^\kappa, (x, w), \mathsf{st}_\mathsf{R}, \mathsf{cds}_2), \mathsf{st}_\mathcal{D}) = 1 | (\mathsf{cds}_1, \mathsf{st}_\mathsf{R}) \leftarrow \mathsf{CDS}_1(1^\kappa), (\mathsf{cds}_2, \mathsf{st}_\mathcal{D}) \leftarrow \mathcal{D}_1(1^\kappa)] -$$

$$\Pr[\mathcal{D}_2(\mathsf{Sim}_2(x, \mathsf{st}_{\mathsf{Sim}}), \mathsf{st}_\mathcal{D}) = 1 | (\mathsf{cds}_1, \mathsf{st}_{\mathsf{Sim}}) \leftarrow \mathsf{Sim}_1(1^\kappa), (\mathsf{cds}_2, \mathsf{st}_\mathcal{D}) \leftarrow \mathcal{D}_1(1^\kappa)]\Big| \leq \mathsf{neg}(\kappa).$$

It can be observed that cds_1 and cds_2 are independent of x and hence can be performed offline in a preprocessing phase.

Construction. We denote an $\mathsf{SSP}_{\mathsf{OT}}$ protocol as $\mathsf{SSP}_{\mathsf{OT}} = (\mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(3)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}_{\mathsf{S}})$.

- $\mathsf{CDS}_1(1^\kappa)$: For $i \in [n]$, the receiver computes $\mathsf{cds}_1^i = \mathsf{ot}_1^i = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa)$. The receiver sends $\mathsf{cds}_1 = \{\mathsf{cds}^i\}_{i\in[n]}$ to the sender and stores $\mathsf{st}_{\mathsf{R}} = \perp$ as the internal state.

- $\mathsf{CDS}_2(1^\kappa, \mathsf{cds}_1)$: For $i \in [n]$, the sender performs the following: $(\mathsf{cds}_2^i, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{S}}^i) = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{cds}_1^i)$. The sender sends $\mathsf{cds}_2 = \{\mathsf{cds}_2^i\}_{i\in[n]}$ to the receiver and stores $\mathsf{st}_{\mathsf{S}} = \{\mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{S}}^i\}$ as the internal sender's state.

- $\mathsf{CDS}_3(1^\kappa, (x, w), \mathsf{st}_{\mathsf{R}}, \mathsf{cds}_2)$: The receiver denotes $w = \{w_i\}_{i\in[n]}$. It computes $(\mathsf{cds}_3^i, \mathsf{m}_i') = \mathsf{OT}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, w_i, \mathsf{cds}_2^i)$ for $i \in [n]$. The receiver sends $\mathsf{cds}_3 = \{\mathsf{cds}_3^i\}_{i\in[n]}$ to sender and stores $\mathsf{st}_{\mathsf{R}} = (w, \{\mathsf{m}_i'\}_{i\in[n]})$ as internal state.

- $\mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}), \mathsf{st}_{\mathsf{S}}, \mathsf{cds}_3)$: The sender performs the following:

  1. Computes the following circuit C:

  $$\begin{aligned} \mathcal{C}(x, w, \mathsf{ptxt}) &= \mathsf{ptxt} \text{ iff } (\mathcal{R}(x, w) == 1) \\ &= 0, \text{ otherwise} \end{aligned}$$

  x is hardcoded in the circuit, $w \in \{0,1\}^n$ and $\mathsf{ptxt} \in \{0,1\}^\ell$ are inputs to the circuit. The sender garbles circuit $\mathcal{C}$ as $(\mathrm{GC}, \mathbf{lab}) \leftarrow \mathsf{Garble.Gb}(1^\kappa, \mathcal{C})$. The computes

  2. For $i \in [n]$, it computes $(\mathsf{m}_0^i, \mathsf{m}_1^i) = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}_{\mathsf{S}}(\mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{S}}^i, \mathsf{cds}_3^i)$.
  3. Parse $\mathbf{lab} = \{\mathsf{lab}_i^0, \mathsf{lab}_i^1\}_{i\in[n+\ell]}$. For $i \in [n], \alpha \in \{0,1\}$, the sender computes $y_i^\alpha = m_i^\alpha \oplus \mathsf{lab}_i^\alpha$. Set $\mathbf{y} = \{y_i^0, y_i^1\}_{i\in[n]}$.
  4. Compute the wire labels corresponding to input $\mathsf{ptxt} \in \{0,1\}^\ell$ as follows ($\mathsf{L}_i = \mathsf{Garble.En}(\mathsf{ptxt}_i, \{\mathsf{lab}_{n+i}^0, \mathsf{lab}_{n+i}^1\})$) for $i \in [\ell]$.

  The sender sends $\mathsf{cds}_4 = (\mathrm{GC}, \mathbf{y}, \{\mathsf{L}_i\}_{i\in[\ell]})$ to the receiver.

- $\mathsf{CDS}_5(\mathsf{st}_{\mathsf{R}}, \mathsf{cds}_4)$: For $i \in [n]$, the receiver computes $\mathsf{lab}_i' = \mathsf{m}_i' \oplus y_i^{w_i}$. The receiver sets $\mathsf{lab}_{n+i}' = \mathsf{L}_i$ for $i \in [\ell]$. The receiver evaluates the garbled circuit to obtain $\mathsf{ptxt}' = \mathsf{Garble.Ev}(\mathrm{GC}, \{\mathsf{lab}_i'\}_{i\in[n+\ell]})$. The receiver outputs ptxt' as the decrypted message.

Correctness. The receiver obtains $\mathsf{lab}_i' = \mathsf{lab}_i^{w_i}$ for $i \in [n]$ from the ith OT protocol corresponding to witness bit w_i. It evaluates the garbled circuit GC to obtain the message $\mathsf{ptxt}' == \mathsf{ptxt}$ if $\mathcal{R}(x, w) = 1$.

Theorem 22 *Assuming $\mathsf{SSP}_{\mathsf{OT}}$ is a four round OT protocol with statistical sender privacy against a malicious receiver and computational receiver privacy against a semi-honest sender, and* Garble *is an information theoretic garbling scheme for NC1 circuits, then CDS is a conditional disclosure of secrets for statements $x \in \mathcal{L}$ which are verifiable by relations $\mathcal{R}(x, \cdot)$ that can be computed by NC1 circuits. Moreover, it provides receiver simulation against a malicious receiver and message indistinguishability against a semi-honest sender.*

Proof. We argue message indistinguishability and receiver simulation as follows.

Message Indistinguishability. We show the following through a sequence of hybrids.

$$\mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}_0), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3^*) \overset{s}{\approx} \mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}_1), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3^*)$$

We present our hybrids and the indistinguishability argument as follows:

1. $\mathtt{Hyb}_0$: $\mathsf{cds}_4 = \mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}_0), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3^*)$ and cds_2 is computed following the honest sender algorithm with ptxt_0 as the sender's input message.

2. $\mathtt{Hyb}_1$: Same as $\mathtt{Hyb}_0$, except the garbled circuit in cds_4 is simulated with output 0 and the simulator obtains simulated wire labels $\widehat{\mathbf{lab}} = \{\widehat{\mathsf{lab}_i}\}_{i\in[n+\ell]}$. The simulator sets $\mathsf{L}_i = \widehat{\mathsf{lab}_{n+i}}$ for $i \in [\ell]$. The simulator sets the sender messages as $y_i^\alpha = \mathsf{m}_i^\alpha \oplus \widehat{\mathsf{lab}_i}$ for $\alpha \in \{0,1\}, i \in [n]$. The two hybrids are statistically indistinguishable due to information theoretic security of the garbling scheme and statistical sender privacy of the SSP_OT.

3. $\mathtt{Hyb}_2$: $\mathsf{cds}_4 = \mathsf{CDS}_4(1^\kappa, (x, \mathsf{ptxt}_1), \mathsf{st}_\mathsf{S}, \mathsf{cds}_3^*)$ and cds_2 is computed following the honest sender algorithm with ptxt_1 as the sender's input message. The two hybrids are statistically indistinguishable due to information theoretic security of the garbling scheme and statistical sender privacy of the SSP_OT.

Receiver simulation. The simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ sets $w = 0^n$ and performs the honest receiver algorithm with w. Indistinguishability follows from the computational receiver privacy of the SSP_OT protocol. □

6.2 Protocol π_{mal}

We compile the 2PC protocol π_{exp} (from Thm. 20), which is secure against unbounded explainable receiver and PPT explainable sender, to be secure against malicious corruptions. Our protocol π_{mal} can be found below and the security is summarized in Thm. 23. High level overview can be found in Sec. 2.1.

Construction. The receiver R has input A and sender S has input B. We present our compiler $\pi_{\mathsf{mal}} = (\mathsf{R}_1, \mathsf{S}_1, \mathsf{R}_2, \mathsf{S}_2, \mathsf{R}_3, \mathsf{S}_3)$ as follows:

- $\mathsf{R}_1(1^\kappa, \mathsf{A})$: The receiver performs the following:
  1. Sample $r_\mathsf{R} \leftarrow \{0,1\}^*$ and compute $\pi^1_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_1(\mathsf{A}; r_\mathsf{R})$ according to the explainable protocol.
  2. Set $(z_1, \mathsf{st}_{\mathsf{ZKP},\mathsf{P}}) \leftarrow \mathsf{ZKP}.\mathsf{P}(1^\kappa)$ and $(z_1', \mathsf{st}_{\mathsf{SZK},\mathsf{V}}) \leftarrow \mathsf{SZK}.\mathsf{V}_1(1^\kappa)$ as the first messages of the ZK proof with R as prover, and SZK argument with R as verifier, respectively.

3. Set $(\mathsf{cds}_1, \mathsf{st}_{\mathsf{CDS,R}}) = \mathsf{CDS.CDS}_1(1^\kappa)$ as the first message of the CDS scheme as receiver.
4. Send $\pi^1_{\mathsf{mal}} = (\pi^1_{\mathsf{exp}}, z_1, z'_1, \mathsf{cds}_1)$.
5. Store $\mathsf{st}_{\mathsf{R}} = (\mathsf{A}, r_{\mathsf{R}}, \mathsf{st}_{\mathsf{ZKP,P}}, \mathsf{st}_{\mathsf{SZK,V}}, \mathsf{st}_{\mathsf{CDS,R}})$.

- $\mathsf{S}_1(1^\kappa, \mathsf{B}, \pi^1_{\mathsf{mal}})$: The sender performs the following:
  1. Sample $r_{\mathsf{S}} \leftarrow \{0,1\}^*$ and set $\pi^2_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{S}_1(\pi^1_{\mathsf{exp}}, \mathsf{B}; r_{\mathsf{S}})$ according to the explainable protocol.
  2. Set $(z_2, \mathsf{st}_{\mathsf{ZKP,V}}) \leftarrow \mathsf{ZKP.V}_1(z_1, 1^\kappa)$, $(z'_2, \mathsf{st}_{\mathsf{SZK,P}}) \leftarrow \mathsf{SZK.P}_1(z'_1)$ as the second message of the ZKPproof with S as verifier, and SZK argument with sender as prover, respectively.
  3. Sample $r^{\mathsf{CDS}}_{\mathsf{S}} \leftarrow \{0,1\}^*$ and compute $(\mathsf{cds}_2, \mathsf{st}_{\mathsf{CDS,S}}) = \mathsf{CDS.CDS}_2(r^{\mathsf{CDS}}_{\mathsf{S}}, \mathsf{cds}_1)$ as the second message of the CDS scheme as sender.
  4. Send $\pi^2_{\mathsf{mal}} = (\pi^2_{\mathsf{exp}}, z_2, z'_2, \mathsf{cds}_2)$.
  5. Store $\mathsf{st}_{\mathsf{S}} = (\mathsf{B}, r_{\mathsf{S}}, \mathsf{st}_{\mathsf{ZKP,V}}, \mathsf{st}_{\mathsf{SZK,P}}, \mathsf{st}_{\mathsf{CDS,S}})$.
- $\mathsf{R}_2(\mathsf{st}_{\mathsf{R}}, \pi^2_{\mathsf{mal}})$: The receiver performs the following:
  1. Compute $\pi^3_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_2(\pi^2_{\mathsf{exp}}, \mathsf{A}; r_{\mathsf{R}})$. Set statement $x_{\mathsf{CDS}} = (\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}})$ and witness $w_{\mathsf{CDS}} = (\mathsf{A}, r_{\mathsf{R}}, \mathsf{ldp})$ where ldp is a low-depth proof of
  $$(\pi^1_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_1(\mathsf{A}; r_{\mathsf{R}}) \wedge \pi^3_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_2(\pi^2_{\mathsf{exp}}, \mathsf{A}; r_{\mathsf{R}})).$$
  2. Compute $(\mathsf{cds}_3, \mathsf{st}_{\mathsf{CDS,R}}) \leftarrow \mathsf{CDS.CDS}_3(1^\kappa, (x_{\mathsf{CDS}}, w_{\mathsf{CDS}}), \mathsf{st}_{\mathsf{CDS,R}}, \mathsf{cds}_2)$.
  3. Compute $(z_3, \mathsf{st}_{\mathsf{ZKP,P}}) \leftarrow \mathsf{ZKP.P}_2(z_2, \mathsf{st}_{\mathsf{ZKP,P}})$ and $(z'_3, \mathsf{st}_{\mathsf{SZK,V}}) \leftarrow \mathsf{SZK.V}_2(z'_2, \mathsf{st}_{\mathsf{SZK,V}})$.
  4. Send $\pi^3_{\mathsf{mal}} = (\pi^3_{\mathsf{exp}}, z_3, z'_3, \mathsf{cds}_3)$.
  5. Update $\mathsf{st}_{\mathsf{R}} = (\mathsf{A}, r_{\mathsf{R}}, \mathsf{st}_{\mathsf{ZKP,P}}, \mathsf{st}_{\mathsf{SZK,V}}, \mathsf{st}_{\mathsf{CDS,R}})$.
- $\mathsf{S}_2(\mathsf{st}_{\mathsf{S}}, \pi^3_{\mathsf{mal}})$: The sender performs the following:
  1. Set $\pi^4_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{S}_2(\pi^3_{\mathsf{exp}}, \mathsf{B}; r_{\mathsf{S}})$.
  2. Set statement $x_{\mathsf{CDS}} = (\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}})$. Compute CDS response
  $$\mathsf{cds}_4 \leftarrow \mathsf{CDS.CDS}_4(r^{\mathsf{CDS}}_{\mathsf{S}}, (x_{\mathsf{CDS}}, \pi^4_{\mathsf{exp}}), \mathsf{st}_{\mathsf{CDS,S}}, \mathsf{cds}_3).$$
  3. Compute $(z_4, \mathsf{st}_{\mathsf{ZKP,V}}) \leftarrow \mathsf{ZKP.V}_2(z_3, \mathsf{st}_{\mathsf{ZKP,V}})$.
  4. Set the statement $x_{\mathsf{SZK}} = (\mathsf{cds}^1, \mathsf{cds}_2, \mathsf{cds}_3, \mathsf{cds}_4, x_{\mathsf{CDS}})$ for witness $w_{\mathsf{SZK}} = (\mathsf{B}, r^{\mathsf{CDS}}_{\mathsf{S}}, r_{\mathsf{S}}, \pi^4_{\mathsf{exp}})$ and set $z'_4 \leftarrow \mathsf{SZK.P}_2(z'_3, x_{\mathsf{SZK}}, \mathsf{st}_{\mathsf{SZK,P}})$.
  5. Send $\pi^4_{\mathsf{mal}} = (\mathsf{cds}_4, z_4, z'_4)$.
  6. Update $\mathsf{st}_{\mathsf{S}} = (\mathsf{B}, r_{\mathsf{S}}, \mathsf{st}_{\mathsf{ZKP,V}}, \mathsf{st}_{\mathsf{SZK,P}})$
- $\mathsf{R}_3(\mathsf{st}_{\mathsf{R}}, \pi^4_{\mathsf{mal}})$: The receiver performs the following:
  1. Set the statement as $x_{\mathsf{SZK}} = (\mathsf{cds}^1, \mathsf{cds}_2, \mathsf{cds}_3, \mathsf{cds}_4, x_{\mathsf{CDS}})$. The receiver aborts if the verification fails as $\mathsf{SZK.V}_3(z'_4, x_{\mathsf{SZK}}, \mathsf{st}_{\mathsf{SZK,V}}) = 0$. Otherwise, decrypt $\pi^4_{\mathsf{exp}} = \mathsf{CDS.CDS}_5(\mathsf{st}_{\mathsf{CDS,R}}, \mathsf{cds}_4)$ and compute the final message as $(\pi^5_{\mathsf{exp}}, \mathsf{out}) = \pi_{\mathsf{exp}}.\mathsf{R}_3(\pi^4_{\mathsf{exp}}, \mathsf{A}; r_{\mathsf{R}})$.

2. Set $x_{\mathsf{ZKP}} = (\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}}, \pi^4_{\mathsf{exp}}, \pi^5_{\mathsf{exp}})$, $w_{\mathsf{ZKP}} = (\mathsf{A}, r_{\mathsf{R}})$ and compute the ZKP proof as $z_5 = \mathsf{ZKP.P}_3(z_4, x_{\mathsf{ZKP}}, \mathsf{st}_{\mathsf{ZKP,P}})$.
3. Send $\pi^5_{\mathsf{mal}} = (\pi^5_{\mathsf{exp}}, z_5)$ to the sender and output out.

- $\mathsf{S}_3(\mathsf{st}_{\mathsf{S}}, \pi^5_{\mathsf{mal}})$: Set statement $x_{\mathsf{ZKP}} = (\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}}, \pi^4_{\mathsf{exp}}, \pi^5_{\mathsf{exp}})$ for ZKP proof. If $\mathsf{ZKP.V}_3(z_5, x_{\mathsf{ZKP}}, \mathsf{st}_{\mathsf{ZKP,V}}) == 0$ then abort. Else, output $\pi_{\mathsf{exp}}.\mathsf{S}_3(\pi^5_{\mathsf{exp}}, \mathsf{B}; r_{\mathsf{S}})$.

We denote the statement for the CDS as follows:

$$\mathsf{L}_{\mathsf{CDS}} = \{(\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}}) : \exists(\mathsf{A}, r_{\mathsf{R}}, \mathsf{ldp}) \text{ s.t. } \mathsf{ldp} \text{ is a low depth proof of}$$

$$\pi^1_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_1(\mathsf{A}; r_{\mathsf{R}}) \wedge \pi^3_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_2(\pi^2_{\mathsf{exp}}, \mathsf{A}; r_{\mathsf{R}})\}$$

The SZK statement proven by the sender is as follows:

$$\mathsf{L}_{\mathsf{SZK}} = \{(\mathsf{cds}_1, \mathsf{cds}_2, \mathsf{cds}_3, \mathsf{cds}_4, x_{\mathsf{CDS}}) : \exists(\mathsf{B}, r^{\mathsf{CDS}}_{\mathsf{S}}, r_{\mathsf{S}}, \pi^4_{\mathsf{exp}}) \text{ s.t. } \pi^2_{\mathsf{exp}} = \mathsf{S}_1(\pi^1_{\mathsf{exp}}, \mathsf{B}; r_{\mathsf{S}}) \wedge$$

$$(\mathsf{cds}_2, \mathsf{st}_{\mathsf{CDS,S}}) = \mathsf{CDS.CDS}_2(r^{\mathsf{CDS}}_{\mathsf{S}}) \wedge \mathsf{cds}_4 = \mathsf{CDS.CDS}_4(r^{\mathsf{CDS}}_{\mathsf{S}}, (x_{\mathsf{CDS}}, \pi^4_{\mathsf{exp}}), \mathsf{st}_{\mathsf{CDS,S}}, \mathsf{cds}_3)\}.$$

We denote the ZKP statement proven by the receiver as follows:

$$\mathsf{L}_{\mathsf{ZKP}} = \{(\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}}, \pi^4_{\mathsf{exp}}, \pi^5_{\mathsf{exp}})\exists(\mathsf{A}, r_{\mathsf{R}}) \text{ s.t. } \pi^1_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_1(\mathsf{A}; r_{\mathsf{R}})$$

$$\wedge\pi^3_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_2(\pi^2_{\mathsf{exp}}, \mathsf{A}; r_{\mathsf{R}}) \wedge \pi^5_{\mathsf{exp}} = \pi_{\mathsf{exp}}.\mathsf{R}_3(\pi^4_{\mathsf{exp}}, \mathsf{A}; r_{\mathsf{R}})\}.$$

Theorem 23 *Assuming the following holds:*

1. *Four round delayed-input adaptive statistical zero-knowledge arguments of knowledge* $\mathsf{SZK} = (V_1, P_1, V_2, P_2, V_3)$ *with adaptive soundness,*
2. *Five round delayed-input adaptive computational zero-knowledge proofs* $ZKP = (P_1, V_1, P_2, V_2, P_3, V_3)$ *with adaptive soundness,*
3. *Four round statistical Conditional Disclosure of Secrets* $CDS = (CDS_1, CDS_2, CDS_3, CDS_4, CDS_5)$ *for NP relations verifiable by NC1 circuits with two rounds of preprocessing phase and two rounds of input-dependent phase,*
4. *Five round robust two-party secure computation protocol* $\pi_{exp}=(R_1, S_1, R_2, S_2, R_3, S_3)$ *against unbounded explainable receiver and PPT explainable sender*

there exists a robust 5-round secure two-party computation protocol $\pi_{mal}=(R_1, S_1, R_2, S_2, R_3, S_3)$ *with black-box simulation against unbounded malicious receivers and PPT malicious senders, where the receiver obtains its output at the end of fourth round and the sender obtains its output at the end of the fifth round.*

Proof. We first consider the case where the receiver is corrupt and then we focus on the corruption of the sender.

Receiver is corrupt. We present our simulation algorithm against a corrupt receiver R^* as follows.

- $\mathsf{R}_1^*(1^\kappa)$: The corrupt receiver sends $\pi_{\mathsf{mal}}^1 = (\pi_{\mathsf{exp}}^1, z_1, z_1', \mathsf{cds}_1)$
- $\mathsf{S}_1(1^\kappa, \pi_{\mathsf{mal}}^1)$: The sender performs the following:
  1. Invoke $\mathsf{Sim}_{\mathsf{exp}}^{\mathsf{S}}(\pi_{\mathsf{exp}}^1)$ to obtain π_{exp}^2.
  2. Set $(z_2, \mathsf{st}_{\mathsf{ZKP,V}}) \leftarrow \mathsf{ZKP.V}_1(z_1, 1^\kappa)$ as the second message of the ZKPproof with S as verifier. Set $z_2' \leftarrow \mathsf{SZK.Sim}_{\mathsf{SZK}}(z_1')$ as the simulated SZK argument with sender as prover.
  3. Set $(\mathsf{cds}_2, \mathsf{st}_{\mathsf{CDS,S}}) = \mathsf{CDS.CDS}_2(1^\kappa)$ as the second message of the CDS scheme as sender.
  4. Send $\pi_{\mathsf{mal}}^2 = (\pi_{\mathsf{exp}}^2, z_2, z_2', \mathsf{cds}_2)$.
  5. Store $\mathsf{st}_{\mathsf{S}} = (\mathsf{st}_{\mathsf{ZKP,V}}, \mathsf{st}_{\mathsf{CDS,S}})$.
- $\mathsf{R}_2^*(\pi_{\mathsf{mal}}^2)$: The corrupt receiver sends $\pi_{\mathsf{mal}}^3 = (\pi_{\mathsf{exp}}^3, z_3, z_3', \mathsf{cds}_3)$.
- $\mathsf{S}_2(\mathsf{st}_{\mathsf{S}}, \pi_{\mathsf{mal}}^3)$: The sender performs the following:
  1. Invoke $\mathsf{Sim}_{\mathsf{exp}}^{\mathsf{S}}(\pi_{\mathsf{exp}}^3)$ to obtain π_{exp}^4.
  2. Set statement $x_{\mathsf{CDS}} = (\pi_{\mathsf{exp}}^1, \pi_{\mathsf{exp}}^2, \pi_{\mathsf{exp}}^3)$. Compute CDS response
  $$\mathsf{cds}_4 \leftarrow \mathsf{CDS.CDS}_4(1^\kappa, (x_{\mathsf{CDS}}, \pi_{\mathsf{exp}}^4), \mathsf{st}_{\mathsf{CDS,S}}, \mathsf{cds}_3).$$
  3. Compute $(z_4, \mathsf{st}_{\mathsf{ZKP,V}}) \leftarrow \mathsf{ZKP.V}_2(z_3, \mathsf{st}_{\mathsf{ZKP,V}})$.
  4. Set the statement $x_{\mathsf{SZK}} = (\pi_{\mathsf{exp}}^2, \mathsf{cds}_4)$ and obtain the simulated SZK argument as set $z_4' \leftarrow \mathsf{SZK.Sim}_{\mathsf{SZK}}(z_3', x_{\mathsf{SZK}})$.
  5. Send $\pi_{\mathsf{mal}}^4 = (\mathsf{cds}_4, z_4, z_4')$.
  6. Update $\mathsf{st}_{\mathsf{S}} = \mathsf{st}_{\mathsf{SZK,P}}$.
- $\mathsf{R}_3^*(\pi_{\mathsf{mal}}^4)$: The corrupt receiver sends $\pi_{\mathsf{mal}}^5 = (\pi_{\mathsf{exp}}^5, z_5)$.
- $\mathsf{S}_3(\mathsf{st}_{\mathsf{S}}, \pi_{\mathsf{mal}}^5)$: If $\mathsf{ZKP.V}_3(z_5, x_{\mathsf{ZKP}}, \mathsf{st}_{\mathsf{ZKP,V}}) == 0$ then abort. Else, invoke the ideal functionality to release the output to honest party.

We present our hybrids and the indistinguishability argument as follows:

1. $\mathtt{Hyb}_0$: Real world execution of the protocol, where the simulator runs the honest sender algorithm with input B, alongwith the output of the honest sender.
2. $\mathtt{Hyb}_1$: Same as $\mathtt{Hyb}_0$, except the simulator generates (z_2', z_4') by invoking the statistical ZK simulator $\mathsf{Sim}_{\mathsf{SZK}}$. The view also includes the output of the honest sender. The two hybrids are statistically indistinguishable due to statistical zero knowledge of SZK.
3. $\mathtt{Hyb}_2$: This is the ideal world execution of the protocol alongwith the output of the honest sender. The only difference between the two hybrids is based on the execution of π_{exp}. In $\mathtt{Hyb}_1$, it is the real world execution of π_{exp} with the honest party's input and in $\mathtt{Hyb}_2$ it is the simulated execution of π_{exp} against a malicious receiver. The two

hybrids are statistically indistinguishable if R^* is explainable. If R^* is not explainable then the statistical soundness of ZKP ensures that the sender aborts at the end of fifth round. If R^* behaves maliciously in the first three rounds of the protocol then the robustness of π_{exp} and (statistical) message indistinguishability of CDS ensures that the two hybrids are statistically indistinguishable.

Sender is corrupt. We present our simulation algorithm against a corrupt sender S^* as follows.

- $\mathsf{R}_1(1^\kappa)$: The simulated receiver performs the following:
  1. Invoke $\mathsf{Sim}^{\mathsf{R}}_{\mathsf{exp}}(1^\kappa)$ to obtain π^1_{exp}.
  2. Obtain $z_1 \leftarrow \mathsf{ZKP.Sim}_{\mathsf{ZKP}}(1^\kappa)$ and $z'_1 \leftarrow \mathsf{SZK.Ext}_{\mathsf{SZK}}(1^\kappa)$ as the first messages of the ZK proof and SZK respectively.
  3. Set $\mathsf{cds}_1 = \mathsf{CDS.Sim}_1(1^\kappa)$ as the first message of the CDS scheme as receiver.
  4. Send $\pi^1_{\mathsf{mal}} = (\pi^1_{\mathsf{exp}}, z_1, z'_1, \mathsf{cds}_1)$.
  5. Store $\mathsf{st}_{\mathsf{R}} = \perp$.
- $\mathsf{S}^*_1(1^\kappa, \pi^1_{\mathsf{mal}})$: The corrupt sender sends $\pi^2_{\mathsf{mal}} = (\pi^2_{\mathsf{exp}}, z_2, z'_2, \mathsf{cds}_2)$.
- $\mathsf{R}_2(\mathsf{st}_{\mathsf{R}}, \pi^2_{\mathsf{mal}})$: The simulated receiver performs the following:
  1. Invoke $\mathsf{Sim}^{\mathsf{R}}_{\mathsf{exp}}(\pi^2_{\mathsf{exp}})$ to obtain π^3_{exp}. Set statement $x_{\mathsf{CDS}} = (\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}})$ following honest receiver algorithm.
  2. Compute $\mathsf{cds}_3 \leftarrow \mathsf{CDS.Sim}_2(1^\kappa, x_{\mathsf{CDS}}, \mathsf{cds}_2)$. Set $z_3 \leftarrow \mathsf{ZKP.Sim}_{\mathsf{ZKP}}(z_2)$ and $z'_3 \leftarrow \mathsf{SZK.Ext}_{\mathsf{SZK}}(z'_2)$.
  3. Send $\pi^3_{\mathsf{mal}} = (\pi^3_{\mathsf{exp}}, z_3, z'_3, \mathsf{cds}_3)$.
- $\mathsf{S}^*_2(\pi^3_{\mathsf{mal}})$: The corrupt sender sends $\pi^4_{\mathsf{mal}} = (\mathsf{cds}_4, z_4, z'_4)$.
- $\mathsf{R}_3(\mathsf{st}_{\mathsf{R}}, \pi^4_{\mathsf{mal}})$: The simulated receiver performs the following:
  1. Set the statement as $x_{\mathsf{SZK}} = (\pi^2_{\mathsf{exp}}, \mathsf{cds}_4)$. The simulated receiver computes $\pi^4_{\mathsf{exp}} = \mathsf{SZK.Ext}_{\mathsf{SZK}}(z'_4)$. Abort if $\pi^4_{\mathsf{exp}} = \perp$.
  2. Compute $\pi^5_{\mathsf{exp}} = \mathsf{Sim}^{\mathsf{R}}_{\mathsf{exp}}(\pi^4_{\mathsf{exp}})$.
  3. Set $x_{\mathsf{ZKP}} = (\pi^1_{\mathsf{exp}}, \pi^3_{\mathsf{exp}}, \pi^5_{\mathsf{exp}})$ and compute the ZKP simulated proof as $z_5 = \mathsf{ZKP.Sim}_{\mathsf{ZKP}}(z_4, x_{\mathsf{ZKP}})$.
  4. Send $\pi^5_{\mathsf{mal}} = (\pi^5_{\mathsf{exp}}, z_5)$ to the sender.
- $\mathsf{S}^*_3(\pi^5_{\mathsf{mal}})$: The corrupt sender performs its own adversarial algorithm.

We present our hybrids and the indistinguishability argument as follows:

1. $\mathtt{Hyb}_0$: Real world execution of the protocol where the simulator runs the honest receiver algorithm with input A, alongwith the output of the honest receiver.

2. $\mathtt{Hyb}_1$: Same as $\mathtt{Hyb}_0$, except the simulated receiver generates (z_1, z_3, z_5) are generated by invoking the ZK simulator of ZKP. The view also includes the output of the honest receiver. The two hybrids are computationally indistinguishable due to computational zero knowledge of ZKP.

3. $\mathtt{Hyb}_2$: Same as $\mathtt{Hyb}_1$, except the simulated receiver generates the CDS messages $(\mathsf{cds}_1, \mathsf{cds}_3)$ by invoking the CDS simulator. The view also includes the output of the honest receiver. The two hybrids are computationally indistinguishable due to receiver simulation of CDS.

4. $\mathtt{Hyb}_3$: This is the ideal world execution of the protocol together with the output of the honest receiver. The only difference between the two hybrids is based on the execution of π_{exp}. In $\mathtt{Hyb}_2$ it is the real world execution of π_{exp} with the honest party's input and in $\mathtt{Hyb}_2$ it is the simulated execution of π_{exp} against a malicious sender. The two hybrids are computationally indistinguishable whenever the corrupt sender in π_{exp} is explainable. When the sender is not explainable, the robustness of π_{exp} ensures that the first three messages $(\pi^1_{\mathsf{exp}}, \pi^2_{\mathsf{exp}}, \pi^3_{\mathsf{exp}})$ are indistinguishable in real world and ideal world executions, even against malicious adversaries, these hybrids are indistinguishable for the first three rounds. The soundness of SZK ensures that in both hybrids the sender algorithm aborts at the end of fourth round if the sender's messages $(\pi^2_{\mathsf{exp}}, \pi^4_{\mathsf{exp}})$ are not explainable. Thus, a sender (whose messages are not explainable) distinguishes between the two hybrids by breaking computational soundness of SZK.

□

7 Instantiations of eOT

We instantiate $\mathsf{eOT}$from CDH, reciprocal CSIDH assumption and two-round $\mathsf{SSP}_{\mathsf{OT}}$ as follows.

7.1 CDH-based Instantiation

We define our elementary OT protocol $\mathsf{eOT} = (\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{OT}_{\mathsf{R}})$ as a tuple of four algorithms defined as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$: The sender samples $Q \leftarrow G$. The sender sends $\mathsf{ot}_1 = Q$ as the OT sender message.

- $\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$: The receiver performs the following with input choice bit γ as follows:

  - Sample $\mathsf{sk} \leftarrow \mathbb{Z}_q$.
  - Set $\mathsf{pk}_\gamma = g^{\mathsf{sk}}$ and set $\mathsf{pk}_{1-\gamma} = \frac{Q}{\mathsf{pk}_\gamma}$.

  The receiver sends $\mathsf{ot}_2 = \mathsf{pk}_0$ as the OT receiver message and sets $\mathsf{st}_{\mathsf{R}} = (\gamma, \mathsf{sk})$.

- $\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2)$: The sender computes following:

- Generate $\mathsf{pk}_1 = \frac{Q}{\mathsf{pk}_0}$.
- Sample $r \leftarrow \mathbb{Z}_q$. Compute $R = g^r$.
- Compute $\mathsf{m}_0 = \mathsf{pk}_0^r$ and $\mathsf{m}_1 = \mathsf{pk}_1^r$.

The sender sends $\mathsf{ot}_3 = R$ as the OT sender message and outputs $(\mathsf{m}_0, \mathsf{m}_1)$ as the output.

- $\mathsf{OT}_\mathsf{R}(\mathsf{st}_\mathsf{R}, \mathsf{ot}_3)$: The receiver computes $\mathsf{m}_\gamma = R^{\mathsf{sk}}$ and outputs m_γ.

Correctness. The sender outputs $(\mathsf{m}_0, \mathsf{m}_1)$. The receiver outputs $\mathsf{m}_\gamma = R^{\mathsf{sk}} = g^{r\mathsf{sk}} = \mathsf{pk}_\gamma^r$ corresponding to bit γ.

Lemma 24 *The above protocol satisfies perfect receiver's elementary security.*

Proof. The distribution of pk_0 is randomly distributed over G irrespective of the value of γ. □

Lemma 25 *The above protocol satisfies computational sender's elementary security based on the CDH assumption.*

Proof. Let $\mathcal{A}$ be an adversary breaking sender privacy of the above OT protocol, then we build an adversary $\mathcal{B}$ breaking the CDH assumption. Recall that $\mathcal{A}$ receives $Q = g^q$ from the sender (for an uniformly sampled $q \leftarrow \mathbb{Z}_q$), sends pk_0 to the sender, receives $R = g^r$ from the sender (for an uniformly sampled $r \leftarrow \mathbb{Z}_q$) and wins if it outputs $\mathsf{m}_0 = \mathsf{pk}_0^r$ and $\mathsf{m}_1 = \mathsf{pk}_1^r$. The CDH adversary $\mathcal{B}$ (acting as the sender) receives $(g, X = g^x, Y = g^y)$ as the CDH challenge. $\mathcal{B}$ sets $Q = X$ and sends it to $\mathcal{A}$. Upon receiving pk_0 from $\mathcal{A}$, $\mathcal{B}$ sends $R = Y$ to $\mathcal{A}$. If $\mathcal{A}$ succeeds then it outputs $\mathsf{m}_0 = \mathsf{pk}_0^y$ and $\mathsf{m}_1 = \mathsf{pk}_1^y$. $\mathcal{B}$ outputs $\mathsf{m}_0 \cdot \mathsf{m}_1$ to the CDH challenger. Recall that $\mathsf{pk}_0 \cdot \mathsf{pk}_1 = Q = X$. If $\mathcal{A}$ succeeds then $\mathcal{B}$ breaks CDH since the following holds:

$$\mathsf{m}_0 \cdot \mathsf{m}_1 = \mathsf{pk}_0^y \cdot \mathsf{pk}_1^y = (\mathsf{pk}_0 \cdot \mathsf{pk}_1)^y = X^y$$

□

7.2 Reciprocal CSIDH-based Instantiation

We recall the reciprocal CSIDH assumption and provide an abstraction of the reciprocal CSIDH assumption in the effective group actions (EGA) framework.

Reciprocal EGA and Reciprocal CSIDH. The OT protocol of Lai et al. [LGdSG21] is based on the reciprocal CSIDH assumption. This assumption is known to be quantum-equivalent to the computational CSIDH assumption, and does not have an analogue in the Diffie-Hellman setting. The construction of Lai et al. relies on crucially on the *quadratic twist* of an elliptic curve, which can be computed efficiently in the CSIDH setting. In this section, we present an abstraction of the quadratic twist and the reciprocal CSIDH assumption in the framework of (R)EGA (Restricted Effective Group Action). In particular, our abstraction

captures all of the properties of quadratic twist and its associated hardness assumptions used by Lai et al (see [LGdSG21] for more details on the quadratic twist and its efficient computation in the CSIDH setting). For the ease of exposition, we present below some preliminary background material on cryptographic group actions and (restricted) effective group actions.

Cryptographic Group Actions. We recall some definitions related to cryptographic group actions from [ADMP20], which provided a framework to construct cryptographic primitives from certain isogeny-based assumptions (e.g., variants of the CSIDH assumption [CLM+18, BKV19]). We begin by describing some notations that we use in the rest of this section. We use $(G, X, \star)$ to denote a group action $\star : G \times X \to X$. Throughout the section, we will assume that group actions are abelian and *regular*, i.e., both free and transitive (which is the case for CSIDH-style group actions). Note that for regular group actions, we have $|G| = |X|$. Thus, if a group action is regular, then for any $x \in X$, the map $f_x : g \to g \star x$ defines a bijection between G and X.

Effective Group Action. We recall the definition of an effective group action (EGA) from [ADMP20]. In a nutshell, an effective group action allows us to do certain computations over G efficiently (e.g., group operation, inversion, and sampling uniformly), and there is an efficient procedure to compute the action of any group element on any set element. As pointed out by [ADMP20], the CSIDH-style assumption in [BKV19] (called "CSI-FiSh") is an instance of effective group action. We refer to [CLM+18, BKV19, ADMP20] for more details on distributional properties of such group actions.

Definition 26 (Effective Group Action) *A group action $(G, X, \star)$ is* effective *if it satisfies the following properties:*

1. *The group G is finite and there exist efficient (PPT) algorithms for:*
   (a) *Membership testing (deciding whether a binary string represents a group element).*
   (b) *Equality testing and sampling uniformly in G.*
   (c) *Group operation and computing inverse of any element in G.*
2. *The set X is finite and there exist efficient algorithms for:*
   (a) *Membership testing (to check if a string represents a valid set element),*
   (b) *Unique representation (there is a canonical representation for any set element $x \in X$).*
3. *There exists a distinguished element $x_0 \in X$ with known representation.*
4. *There exists an efficient algorithm that given any $g \in G$ and any $x \in X$, outputs $g \star x$.*

Restricted Effective Group Action. The authors of [ADMP20] pointed out that while EGA is a useful abstraction, sometimes it is too powerful in comparison to what is achievable in practice. A *Restricted Effective Group Action* (REGA) is a weakening of EGA, where we can only evaluate the action of a generating set of small cardinality.

Definition 27 (Restricted Effective Group Action) *Let $(G, X, \star)$ be a group action and let $\mathbf{g} = (g_1, \ldots, g_n)$ be a (not necessarily minimal) generating set for G. The action is said to be* $\mathbf{g}$-restricted effective*, if the following properties are satisfied:*

- *G is finite and $n = \mathsf{poly}(\log(|G|))$.*
- *The set X is finite and there exist efficient algorithms for:*
  1. *Membership testing, i.e., to decide if a bit string represents a valid set element.*
  2. *Unique representation, i.e., to compute a string $\hat{x}$ that canonically represents any given set element $x \in X$.*
- *There exists a distinguished element $x_0 \in X$, called the* origin*, such that its bit-string representation is known.*
- *There exists an efficient algorithm that given any $i \in [n]$ and any bit string representation of $x \in X$, outputs $g_i \star x$ and $g_i^{-1} \star x$.*

Although an REGA is limited to evaluations of the form $g_i \star x$, this is actually enough to evaluate the action of many, and potentially all elements of G without even needing axioms on the effectivity of G. Protocols built on REGA will need to sample elements from G that are statistically close to uniform and for which the group action is efficiently computable. Prior works suggest sampling from a distribution $\mathcal{D}_G$ on the words on $\mathbf{g}$ in the non-abelian case, or from a distribution on vectors in Z^n in the abelian case (e.g., discrete Gaussian distributions [DG19]), which is plausibly sufficient for applications that require group elements to be sampled from distributions statistically close to uniform [DG19].

The Twist Map. Let $(G, X, \star)$ be an EGA (equivalently an REGA) as described above. We define a "twist" as a map $\mathcal{T} : X \to X$ that satisfies the following properties:

- For any $g \in G$ and any $x \in X$ we have $\mathcal{T}(g \star x) = g^{-1} \star \mathcal{T}(x)$.
- For any $x \in X$ and any *uniform* $g \leftarrow G$, we have: $g \star x \approx_s \mathcal{T}(g \star x)$.
- There exists a "twist-invariant" element $x_0 \in X$ such that $\mathcal{T}(x_0) = x_0$.

The Reciprocal EGA Assumption. Given an EGA $(G, X, \star)$, we say that the reciprocal assumption holds if for any security parameter $\kappa \in \mathbb{N}$ and for any PPT adversary $\mathcal{A}$, the following holds with overwhelmingly large probability:

$$\Pr[\mathsf{Expt}^{\mathsf{recEGA}}(\kappa, \mathcal{A}) = 1] < \mathsf{negl}(\kappa),$$

where the experiment $\mathsf{Expt}^{\mathsf{recEGA}}(\kappa, \mathcal{A})$ is as defined in Figure 2.

Experiment $\mathsf{Expt}^{\mathsf{recEGA}}(\kappa, \mathcal{A})$:

1. The challenger generates the description of an EGA $(G, X, \star)$ along with the "twist" map $\mathcal{T} : X \to X$ and a special "twist-invariant" element $x_{\mathcal{T}} \in X$.

2. The challenger then samples $g \leftarrow G$, sets $x = g \star x_{\mathcal{T}}$, and provides to the adversary $\mathcal{A}$ the tuple $(G, X, \star, \mathcal{T}, x_{\mathcal{T}}, x)$.

3. The adversary $\mathcal{A}$ outputs an element $z \in X$.

4. The challenger samples $s \leftarrow G$ and provides to the adversary $\mathcal{A}$ the set element $y = s \star x$.

5. The adversary $\mathcal{A}$ eventually outputs a pair of set elements $(z_0, z_1) \in X \times X$.

6. Output 1 if $(z_0, z_1) = (s \star z, s^{-1} \star z)$. Output 0 otherwise.

Figure 2: The Reciprocal EGA Experiment

Remark 1 *We can similarly define a reciprocal REGA assumption where, in the corresponding experiment, all group elements (more concretely, the group elements g and s) are sampled from a distribution that is statistically close to uniform over the group G.*

Finally, we import the following theorem from [LGdSG21].

Theorem 28 ([LGdSG21]). *Assuming that the reciprocal CSIDH assumption holds, there exists an REGA satisfying the reciprocal REGA assumption.*

Next, we present our protocol. Let x_0 be a publicly known twist-invariant set element such that $\mathcal{T}(x_0) = x_0$. x_0. Our protocol is as follows:

- $\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$: The sender samples $g \leftarrow G$ and computes $x = g \star x_0$. The sender sends $\mathsf{ot}_1 = x$ as the OT sender message.
- $\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$: The receiver samples $r \leftarrow G$ and computes $z \in X$ as follows:
$$z = r \star x \text{ if } \gamma = 0, \quad z = \mathcal{T}(r \star x) \text{ if } \gamma = 1,$$
The receiver sends $\mathsf{ot}_2 = z$ as the OT receiver message and sets $\mathsf{st}_{\mathsf{R}} = (\gamma, r)$
- $\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2)$: The sender samples $s \leftarrow G$ and computes $(y, \mathsf{m}_0, \mathsf{m}_1)$ as follows:
$$y = s \star x, \quad \mathsf{m}_0 = s \star z, \quad \mathsf{m}_1 = s \star \mathcal{T}(z)$$
The sender sends $\mathsf{ot}_3 = y$ as the OT sender message and outputs $(\mathsf{m}_0, \mathsf{m}_1)$ as the output.
- $\mathsf{OT}_{\mathsf{R}}(\mathsf{st}_{\mathsf{R}}, \mathsf{ot}_3)$: The receiver computes $\mathsf{m}_\gamma = r \star y$ and outputs m_γ.

Correctness. The sender outputs $(\mathsf{m}_0, \mathsf{m}_1)$. We argue correctness based on γ as follows. When $(\gamma == 0)$, $\mathsf{m}_0 = s * z = s * r * x = r * s * x = r * y$. When $(\gamma == 1)$, $\mathsf{m}_1 = s * \mathcal{T}(z) = s * \mathcal{T}(\mathcal{T}(r \star x)) = s * r * x = r * s * x = r * y$. Thus, in both cases the receiver outputs the correct message as $\mathsf{m}_\gamma = r * y$ corresponding to bit γ.

Lemma 29 *Given a set element $x \in X$ and a group G, let $G \star X$ be the distribution on X of $g \star x$ for $g \leftarrow G$, and let $\mathcal{T}(G \star X)$ be the distribution on x of $\mathcal{T}(g \star x)$ for $g \leftarrow G$. If $g \star x$ and $\mathcal{T}(g \star x)$ are statistically indistinguishable the above protocol satisfies statistical receiver's elementary security.*

Proof. The distribution of $r \star x$ and $\mathcal{T}(r \star x)$ is statistically close and hence γ remains statistically hidden. □

Lemma 30 *The above protocol (resp., its CSIDH-based instantiation) satisfies computational sender's elementary security based on the reciprocal EGA assumption in $(\star, G, X)$ (resp., the reciprocal CSIDH assumption).*

Proof. Let $\mathcal{A}$ be an adversary breaking sender privacy of the above OT protocol, then we build an adversary $\mathcal{B}$ breaking the reciprocal assumption. When $\mathcal{B}$ receives x from the challenger of the reciprocal EGA game, $\mathcal{B}$ forwards x to $\mathcal{A}$. $\mathcal{A}$ sends z to $\mathcal{B}$ and $\mathcal{B}$ forwards it to the challenger of the reciprocal EGA game. The challenger returns a y and $\mathcal{B}$ forwards it to $\mathcal{A}$. When $\mathcal{A}$ responds with $(\mathsf{m}_0, \mathsf{m}_1)$ to break sender elementary security, $\mathcal{B}$ sends $(\mathsf{m}_0^*, \mathsf{m}_1^*) = (\mathsf{m}_0, \mathcal{T}(\mathsf{m}_1))$ to the challenger of the reciprocal EGA game as its response. If $\mathcal{A}$ succeeds then $\mathcal{B}$ breaks reciprocal EGA assumption since the following holds:

$$\mathsf{m}_0^* = \mathsf{m}_0 = s * z, \quad \mathsf{m}_1^* = \mathcal{T}(\mathsf{m}_1) = \mathcal{T}(s \star \mathcal{T}(z)) = s^{-1} \star z.$$

□

7.3 Instantiation from 2-round $\mathsf{SSP}_{\mathsf{OT}}$

The 3 round SRP elementary OT eOT can be instantiated from a 2 round $\mathsf{SSP}_{\mathsf{OT}} = (\mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}_{\mathsf{R}})$ based on OT reversal techniques [WW06].

- $\mathsf{OT}^{(1)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa)$: The sender samples a random $\alpha \leftarrow \{0,1\}$ and computes the following - $(\mathsf{SSP}_{\mathsf{OT}}.\mathsf{ot}_1, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{R}}) = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(1)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \alpha)$. The sender sends $\mathsf{ot}_1 = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{ot}_1$ as the OT sender message and stores $\mathsf{st}_{\mathsf{S}} = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{R}}$ as the internal state.
- $\mathsf{OT}^{(2)}_{\mathsf{R}\to\mathsf{S}}(1^\kappa, \gamma, \mathsf{ot}_1)$: The receiver computes the following with input choice bit γ :
  - Samples $r \leftarrow \{0,1\}$.
  - Computes $(\mathsf{SSP}_{\mathsf{OT}}.\mathsf{ot}_2, \mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{S}}) = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}^{(2)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, (r, r \oplus \gamma), \mathsf{ot}_1)$.

  The receiver sends $\mathsf{ot}_2 = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{ot}_2$ and stores $\mathsf{st}_{\mathsf{R}} = r$ as the internal state.
- $\mathsf{OT}^{(3)}_{\mathsf{S}\to\mathsf{R}}(1^\kappa, \mathsf{ot}_2)$: The sender computes the following:
  - Computes $a = \mathsf{SSP}_{\mathsf{OT}}.\mathsf{OT}_{\mathsf{R}}(\mathsf{SSP}_{\mathsf{OT}}.\mathsf{st}_{\mathsf{R}}, \mathsf{ot}_2)$.

– Samples $\mathsf{m}_0 \leftarrow \{0,1\}$ and sets $\mathsf{m}_1 = \mathsf{m}_0 \oplus \alpha$.

– Sets $\mathsf{Map} = \mathsf{m}_0 \oplus a$.

The sender sends $\mathsf{ot}_3 = \mathsf{Map}$ and outputs $(\mathsf{m}_0, \mathsf{m}_1)$.

- $\mathsf{OT}_\mathsf{R}(\mathsf{st}_\mathsf{R}, \mathsf{ot}_3)$: The receiver outputs $\mathsf{m}_\gamma = \mathsf{Map} \oplus r$.

The correctness of the above protocol follows from the correctness of the $\mathsf{SSP}_{\mathsf{OT}}$ and the OT reversal trick.

Lemma 31 *If $\mathsf{SSP}_{\mathsf{OT}}$ is a two round statistically sender private oblivious transfer with computational receiver privacy then the above eOT protocol is statistically receiver private with computational sender privacy.*

Proof. The receiver's choice bit γ is statistically hidden in ot_2 since the sender (acting as the receiver in $\mathsf{SSP}_{\mathsf{OT}}$) either obtains r or $r \oplus \gamma$ due to statistical sender privacy of $\mathsf{SSP}_{\mathsf{OT}}$. Similarly, α is computationally hidden from the receiver (acting as the sender in the $\mathsf{SSP}_{\mathsf{OT}}$ protocol) due to computational receiver privacy of $\mathsf{SSP}_{\mathsf{OT}}$. □

Next, we instantiate the two round $\mathsf{SSP}_{\mathsf{OT}}$ protocol based on various assumptions and obtain the following result.

Theorem 32 *Assuming that the Learning with Errors assumption, Decisional Diffie Hellman assumption, Quadratic Residuosity assumption, N^{th} residuosity assumption, Learning parity with Noise + a standard Nisan-Wigderson style derandomization assumption, or decisional CSIDH assumption holds then the above there exists a three round eOT which satisfies statistical receiver privacy and computational sender privacy.*

Proof. The 2 round $\mathsf{SSP}_{\mathsf{OT}}$ protocol can be instantiated from LWE [BD18], DDH [NP01], QR [HK12], N^{th}-residuosity [HK12], LPN+derandomization techniques [BF22], or decisional CSIDH assumptions [ADMP20]. □

Acknowledgments

We thank the anonymous reviewers of IACR TCC 2022 for their helpful comments and suggestions. Pratik Sarkar is supported by NSF Awards 1931714, 1414119, and the DARPA SIEVE program.

References

[ACJ17] Prabhanjan Ananth, Arka Rai Choudhuri, and Abhishek Jain. A new approach to round-optimal secure multiparty computation. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 468–499. Springer, Heidelberg, August 2017.

[ADMP20] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *ASIACRYPT 2020, Part II*, LNCS, pages 411–439. Springer, Heidelberg, December 2020.

[AMPS21] Navid Alamati, Hart Montgomery, Sikhar Patranabis, and Pratik Sarkar. Two-round adaptively secure MPC from isogenies, lpn, or CDH. In *ASIACRYPT 2021*, volume 13091 of *Lecture Notes in Computer Science*, pages 305–334. Springer, 2021.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390. Springer, Heidelberg, November 2018.

[BF22] Nir Bitansky and Sapir Freizeit. Statistically sender-private ot from lpn and derandomization. Cryptology ePrint Archive, Report 2022/185, 2022. https://ia.cr/2022/185.

[BFJ+20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, LNCS, pages 642–667. Springer, Heidelberg, May 2020.

[BGJ+18] Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal MPC. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 459–487. Springer, Heidelberg, August 2018.

[BHP17] Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 645–677. Springer, Heidelberg, November 2017.

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *ACM CCS 2012*, pages 784–796. ACM Press, October 2012.

[BJY97] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 280–305. Springer, Heidelberg, May 1997.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *ASIACRYPT 2019, Part I*, LNCS, pages 227–247. Springer, Heidelberg, December 2019.

[Blu81] Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *CRYPTO'81*, volume ECE Report 82-04, pages 11–15. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., 1981.

[CCG+20] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Round optimal secure multiparty computation from minimal assumptions. In *TCC 2020, Part II*, LNCS, pages 291–319. Springer, Heidelberg, March 2020.

[CCG+21] Arka Rai Choudhuri, Michele Ciampi, Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. Oblivious transfer from trapdoor permutations in minimal rounds. In *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part II*, volume 13043 of *Lecture Notes in Computer Science*, pages 518–549. Springer, 2021.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018.

[CO17] Wutichai Chongchitmate and Rafail Ostrovsky. Circuit-private multi-key FHE. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 241–270. Springer, Heidelberg, March 2017.

[COSV17] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Round-optimal secure two-party computation from trapdoor permutations. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 678–710. Springer, 2017.

[DG19] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Heidelberg, May 2019.

[DGH+20] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, LNCS, pages 768–797. Springer, Heidelberg, May 2020.

[DN07] Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 2007.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In Vincent Rijmen and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, LNCS, pages 668–699. Springer, Heidelberg, May 2020.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996.

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 448–476. Springer, Heidelberg, May 2016.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 1991.

[HHPV18] Shai Halevi, Carmit Hazay, Antigoni Polychroniadou, and Muthuramakrishnan Venkitasubramaniam. Round-optimal secure multi-party computation. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 488–520. Springer, Heidelberg, August 2018.

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology*, 25(1):158–193, January 2012.

[HNO+09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 2009.

[HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 397–429. Springer, Heidelberg, August 2016.

[Kat08] Jonathan Katz. Which languages have 4-round zero-knowledge proofs? In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 73–88. Springer, 2008.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 34–65. Springer, Heidelberg, April / May 2018.

[KM20] Dakshita Khurana and Muhammad Haris Mughees. On statistical security in two-party computation. In *TCC 2020, Part II*, LNCS, pages 532–561. Springer, Heidelberg, March 2020.

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 335–354. Springer, Heidelberg, August 2004.

[Kol05] Vladimir Kolesnikov. Gate evaluation secret sharing and secure one-round two-party computation. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 136–155. Springer, Heidelberg, December 2005.

[LGdSG21] Yi-Fu Lai, Steven D. Galbraith, and Cyprien Delpech de Saint Guilhem. Compact, efficient and uc-secure isogeny-based oblivious transfer. In *EUROCRYPT 2021*, pages 213–241, 2021.

[LP07] Yehuda Lindell and Benny Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In Moni Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 52–78. Springer, Heidelberg, May 2007.

[LP09] Yehuda Lindell and Benny Pinkas. A proof of security of Yao's protocol for two-party computation. *Journal of Cryptology*, 22(2):161–188, April 2009.

[LS91] Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO'90*, volume 537 of *LNCS*, pages 353–365. Springer, Heidelberg, August 1991.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. *IACR Cryptol. ePrint Arch.*, page 279, 2019.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 735–763. Springer, Heidelberg, May 2016.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptol.*, 1991.

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for *NP* using any one-way permutation. *J. Cryptol.*, 1998.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th SODA*, pages 448–457. ACM-SIAM, January 2001.

[OPP14] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 536–553. Springer, Heidelberg, August 2014.

[ORS15] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 339–358. Springer, Heidelberg, August 2015.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd FOCS*, pages 366–375. IEEE Computer Society Press, November 2002.

[WW06] Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 222–232. Springer, Heidelberg, May / June 2006.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.