Statistical Zaps and New Oblivious Transfer Protocols

Vipul Goyal[1], Abhishek Jain[2], Zhengzhong Jin[2], and Giulio Malavolta[3]

[1]Carnegie Mellon University
[2]Johns Hopkins University
[3]UC Berkeley & Carnegie Mellon University

Abstract

We study the problem of achieving *statistical privacy* in interactive proof systems and oblivious transfer – two of the most well studied two-party protocols – when limited rounds of interaction are available.

- **Statistical Zaps:** We give the first construction of statistical Zaps, namely, two-round statistical witness-indistinguishable (WI) protocols with a *public-coin* verifier. Our construction achieves computational soundness based on the quasi-polynomial hardness of learning with errors.
- **Three-Round Statistical Receiver-Private Oblivious Transfer:** We give the first construction of a three-round oblivious transfer (OT) protocol – in the plain model – that achieves statistical privacy for receivers and computational privacy for senders against malicious adversaries, based on *polynomial-time* assumptions. The round-complexity of our protocol is optimal.

We obtain our first result by devising a public-coin approach to compress sigma protocols, without relying on trusted setup. To obtain our second result, we devise a general framework via a new notion of *statistical hash commitments* that may be of independent interest.

1 Introduction

We study the problem of achieving statistical privacy in two-party cryptographic protocols. Statistical privacy is very appealing in cryptography since it guarantees *everlasting security* – even if the adversary is computationally unbounded during the protocol execution and later post-processes the protocol transcript for as long as it wants, it cannot violate the privacy guarantee. For this reason, perhaps unsurprisingly, statistical privacy is typically much harder to achieve than computational privacy. For example, achieving statistical privacy for *both* participants in two-party protocols is impossible in general.

Nevertheless, in many scenarios, "one-sided" statistical privacy is possible to achieve. In other words, it is typically possible to design protocols that guarantee statistical privacy for one participant and computational privacy for the other. In this work, we investigate the possibility of achieving such asymmetric guarantees when *limited* rounds of interaction are available. We narrow the focus of our study on interactive proof systems [GMR85, Bab85] and oblivious transfer [Rab81, EGL85], two of the most well-studied two-party protocols in the cryptography literature.

Statistical Zaps. The notion of witness-indistinguishable (WI) proofs [FS87] allows a prover to convince a verifier about the validity of a statement (say) x in a manner such that the proof does not reveal which one of possibly multiple witnesses that attest to the validity of x was used in the computation. More specifically, if w_1, w_2 are both witnesses for x, then the verifier should not be able to distinguish between an honest prover using w_1 from an honest prover using w_2. Despite offering a weaker privacy guarantee than zero-knowledge (ZK) proofs [GMR85], WI has found wide applications in cryptography. One reason for its appeal is that most known round-complexity lower bounds for ZK do not apply to WI.

The seminal work of Dwork and Naor [DN00] proved that unlike ZK [GO94], WI can be achieved in two rounds, without relying on a trusted setup. They constructed two-round WI protocols with a *public-coin* verifier message, which they termed *Zaps*, from non-interactive zero-knowledge (NIZK) proofs in the common random string model [DMP88, FLS90]. By relying on known constructions of such NIZKs, their methodology can be used to obtain Zaps from quadratic residuosity [DMP88], trapdoor permutations [FLS90] and the decisional linear assumption over bilinear groups [GOS06b]. More recently, Zaps were also constructed based on indistinguishability obfuscation [BP15].

Over the years, Zaps have found numerous applications in cryptography. Part of their appeal is due to the public-coin verifier property which is crucial to many applications. In particular, it implies *public verifiability*, a property which is often used in the design of round-efficient secure multiparty computation protocols (see, e.g., [HHPV18]). Moreover, it also allows for the verifier message to be *reusable* across multiple proofs, a property which is often used, for example, in the design of resettably-secure protocols (see, e.g., [DGS09]).

Remarkably, all known constructions of Zaps (as well as non-interactive WI [BOV03, GOS06a, BP15]) only achieve *computational* WI property. Despite several years of research, the following fundamental question has remained open:

Do there exist statistical Zaps?

In fact, even two-round statistical WI that only satisfy public-verifiability or reusability, in isolation, are not known currently. This is in contrast to NIZKs, which are indeed known with statistical privacy [CCH+19, PS19] or even perfect privacy [GOS06b]. One reason for this disparity is that the methodology of [DN00] for constructing Zaps is not applicable in the statistical case.

The recent work of Kalai, Khurana and Sahai [KKS18] comes close to achieving this goal. They constructed two round statistical WI with *private-coin* verifier message based on two round statistical sender-private oblivious transfer (OT) [NP01, AIR01, Kal05, HK12, BD18]. The use of a private-coin verifier message is, in fact, instrumental to their approach (which builds on [JKKR17, BGI+17]). As such, a different approach is required for constructing statistical Zaps with a public-coin verifier.

Statistical Receiver-Private Oblivious Transfer. An oblivious transfer (OT) [Rab81, EGL85] protocol allows a "sender" to transfer one of its two inputs to a "receiver" without learning which of the inputs was obtained by the receiver. OT is of special importance to the theory and practice of secure computation [Yao86, GMW87] since OT is both necessary and complete [Kil88] for computing general functions.

Nearly two decades ago, the influential works of works of Naor and Pinkas [NP01] and Aiello et. al. [AIR01] constructed two-round OT protocols that achieve game-based security against malicious adversaries in the plain model. An important property of these protocols is that they guarantee *statistical privacy for senders* (and computational privacy for receivers). Subsequent to these works, new constructions of such protocols were proposed based on a variety of assumptions (see, e.g., [Kal05, HK12, BD18]). Over the years, such OT protocols have found many applications such as constructions of two-round (statistical) WI [JKKR17, BGI+17, KKS18], non-malleable commitments [KS17], and more.

A natural question is whether it is possible to construct such OT protocols with a "reverse" guarantee, namely, *statistical privacy for receivers* (and computational privacy for senders). As observed in [KKS18], two rounds are insufficient for this task: statistical receiver privacy implies that there exists different randomness tapes for receiver that explains a fixed receiver message for both input bits 0 and 1. Thus, a non-uniform malicious PPT receiver could simply start a two-round protocol with non-uniform advice that consists of such a message and randomness tapes, and then use both random tapes to learn *both* inputs of the sender, thereby violating sender privacy.

In the same work, [KKS18] also proved that three rounds are sufficient for this task. Namely, they constructed three round statistical receiver-private OT with game-based security against malicious adversaries, in the plain model. However, they achieve this result by relying upon *super-polynomial-time* hardness assumptions. In contrast, two-round statistical sender-private OT protocols are known from a variety of polynomial-time assumptions. This leaves open the following important question:

Does there exist three-round statistical receiver-private OT in the plain model based on polynomial-time assumptions?

1.1 Our Results

In this work, we resolve both of the aforementioned questions in the affirmative.

I. Statistical Zap Arguments. We give the first construction of statistical Zaps with computational soundness, a.k.a. *statistical Zap arguments*. The soundness of our protocol is based on the quasi-polynomial hardness of the learning with errors (LWE) assumption. While we focus on achieving statistical privacy, we note that our construction, in fact, also yields the first computational Zap argument system based on (quasi-polynomial) LWE.

Theorem 1 (Informal)**.** *Assuming quasi-polynomial LWE, there exists a statistical Zap argument system.*

In order to obtain our result, we depart from prior approaches for constructing Zaps. Specifically, our approach combines the recent statistical NIZK arguments of Peikert and Shiehian [PS19] in a non-black-box manner with a two-round *public-coin* statistically-hiding extractable commitment scheme (see Section 4.1). Previously, such a commitment scheme in the private-coin setting was constructed by [KKS18].

Roughly speaking, while the work of [PS19] (following [CCH$^+$19]) instantiates the Fiat-Shamir methodology [FS87] for compressing sigma protocols [CDS94] into a NIZK using collision-intractable hash (CIH) functions [CGH98], our approach can be seen as a way to compress sigma protocols into statistical Zaps using CIH and two-round public-coin statistically-hiding extractable commitments, without using a trusted setup. Importantly, while prior approaches for compressing sigma protocols into two-round WI [JKKR17, BGI$^+$17, KKS18] lose the public-coin property of the sigma protocol, our approach retains it. We refer the reader to Section 2.1 for more details on our technical approach.

Related work. In a concurrent and independent work, Badrinarayanan et al. [BFJ$^+$20] also construct statistical Zap arguments from quasi-polynomial LWE. In another concurrent and independent work, Lombardi et al. [LVW19] construct computational Zap arguments from quasi-polynomial LWE. In a follow up work, Lombardi et al. [LVW20] construct statistical Zaps with private verifier randomness from quasi-polynomial decisional linear assumption over groups with bilinear maps.

II. Three-Round Statistical Receiver-Private Oblivious Transfer. We devise a general framework for constructing three-round statistical receiver-private OT via a new notion of *statistical hash commitments* (SHC). This notion is inspired by hash proof systems [CS02] that were previously used to design two-round statistical sender-private OT [Kal05, HK12]. Roughly speaking, an SHC scheme is a two-round statistically hiding commitment scheme where the opening verification simply involves an equality check with a hash output (computed w.r.t. a hashing algorithm associated with the scheme).

We devise a generic transformation from any SHC scheme with statistical hiding property to three-round statistical receiver-private OT. The resulting OT scheme achieves game-based security against malicious adversaries in the plain model. For the case of senders, we in fact, achieve a stronger notion of distinguisher-dependent simulation security [DNRS99, JKKR17]. Next, we provide two instantiations of an SHC scheme:

- First, we provide a construction of SHC based on any two-round statistical sender-private OT. Such schemes are known from on a variety of assumptions, including DDH, Quadratic (or N^{th}) Residuosity, and LWE. This yields a new approach for *OT reversal* [WW06] in the context of game-based security, unlike prior works that studied OT reversal in the simulation-based security regime.

- We also provide a construction based on a search assumption, specifically, the computational Diffie-Hellman (CDH) problem. This construction, in fact, achieves *perfect* hiding property.

Putting these together, we obtain the following result:

Theorem 2 (Informal)**.** *Assuming the existence of any two-round statistical sender-private OT (resp., polynomial hardness of CDH), there exists a three-round statistical (resp., perfect) receiver-private OT in the plain model.*

2 Technical Overview

2.1 Statistical Zap Arguments

We now prove a high-level overview of the main ideas underlying our construction of statistical Zaps. Roughly speaking, we devise a strategy to compress sigma protocols into statistical Zaps. While the idea of compressing sigma protocols to two-round WI arguments has been considered before [JKKR17, BGI+17, KKS18], the resulting protocol in these works were inherently private coin as they use oblivious transfer to "hide" the verifier message in the underlying sigma protocol. To obtain a public-coin protocol, we take a different approach.

Our starting point is the recent construction of NIZKs from LWE [PS19, CCH+19] that compresses any "trapdoor" sigma protocol into a NIZK by instantiating the Fiat-Shamir transformation [FS87] in the CRS model. We start by briefly recalling these constructions.

Recent Constructions of NIZKs from LWE. The main tool underlying the constructions of NIZK in [PS19, CCH+19] is the notion of Correlation Intractable Hash (CIH) functions. Roughly speaking, correlation intractability means that for any multi-bit-output circuit f, if we sample a hash function $\mathsf{H}_k(\cdot)$ from the CIH function family, it is hard to find an input x such that $\mathsf{H}_k(x)$ coincides with $f(x)$.

The work of [PS19] construct a NIZK for the Graph Hamiltonian Language[1] starting from a sigma protocol for the same language. Recall that the first round prover message in the

[1] Their construction, in fact, works for any trapdoor sigma protocol.

sigma protocol consists of commitments to some random cycle graphs. Let α denote the cycle graphs. The compression strategy works as follows: first, the prover prepares commitments to α by using a public-key encryption scheme, where the public-key is a part of the CRS setup. Next, the prover computes the verifier's challenge in the sigma protocol by evaluating the CIH function over the first round message, where the CIH key is also fixed by the CRS setup. Given this challenge, the prover finally computes the third round message of the sigma protocol. The NIZK proof simply consists of this transcript.

Roughly speaking, the zero knowledge property of this construction relies on the semantic security of the public key encryption scheme (used to commit α) as well as the programmability of the CIH. Moreover, when the public key is *lossy*, then the NIZK in fact achieves *statistical zero knowledge* property.

The soundness property crucially relies upon the ability to *extract* the values α from the commitments by using the secret key corresponding to the public-key fixed by the CRS, as well as the correlation intractability of the CIH. Specifically, for any instance that is not in the language, given the secret key of the public key encryption, one can extract α from the commitment by decrypting it using the secret key, and then check if α corresponds to cycle graphs or not. Note that this checking procedure can be viewed as a function f. Then, if the malicious prover can find an accepting proof for the false statement, it implies that the output of the function f (with the secret key hardwired) evaluated over first round prover message coincides with the verifier's challenge bits, which are outputted by the CIH function. However, from the correlation intractability of CIH, such a prover shouldn't exist.

Starting Observations. Towards constructing statistical Zaps in the plain model, a naive first idea would be to simply let the verifier generate and send the CRS of the (statistical) NIZK in the first round, and then require the prover to compute and send the NIZK proof based on this CRS in the second round. This attempt, however, fails immediately since the verifier may use the trapdoor corresponding to the CRS (specifically, the secret key corresponding to the public-key encryption) to extract the prover's witness.

One natural idea to address this issue is to replace the public-key encryption scheme with a two-round statistically-hiding commitment scheme. However, while this seems to address witness privacy concerns, it is no longer clear how to argue soundness since the proof of soundness (as discussed above) crucially requires the ability to extract the α values.

Achieving Weak Privacy. In order to devise a solution to the above problems, let us first consider a significantly weaker goal of constructing a two-round protocol that achieves computational soundness but only a very weak form of privacy guarantee, namely, that the verifier can learn the prover's witness with probability at most one-half. Moreover, we do not require the protocol to be public-coin, but only satisfy the weaker property of public verifiability.

To obtain such a protocol, we rely on a 2-round statistical sender-private oblivious transfer protocol in plain model [NP01, Kal05, HK12, BD18]. In such an OT scheme, even if the receiver is malicious, at least one of the sender's messages remains statistically hidden from the receiver. Given such an OT scheme, we construct the desired two-round protocol as follows:

- In the first round, the verifier acts as the OT receiver, and sends a first round OT message with a random input bit b.

- In the second round, the prover prepares a transcript of the sigma protocol in the same manner as in the NIZK construction earlier, with the following key difference: it flips a coin b' and instead of computing the first round prover message as encryptions of α values, it computes OT sender messages where in each message, he uses inputs m_0, m_1, where $m_{b'} = \alpha$ and $m_{1-b'} = \bot$.

With probability one-half, the random bit b of the verifier and the random coin b' of the prover are *different*. In this case, the statistical sender-privacy of the OT ensures that the α values remain hidden from the verifier. As such, the construction satisfies weak privacy, as required.

For computational soundness, consider any instance that is not in the language. Suppose we have an efficient cheating prover that can generate an accepting proof with non-negligible probability. In this case, we can run the cheating prover multiple times to estimate the distribution of the random coin b'. Note that at least one side of the random coin appears with probability no less than half. Without loss of generality, let assume such side is 0. Now we can switch the verifier's random hidden bit b in the first round message of OT to 0. Since the first round message of OT computationally hides b, the efficient cheating prover should not notice the switch, and hence the two random bits coincide with constant probability. However, when the two bits coincide, we can extract α by using the receiver's trapdoor of the OT. This allows us to contradict the correlation intractability of CIH, in the same manner as before.

Finally, note that the verifier does not need to use the randomness of the OT receiver to verify the proof; as such the above construction is publicly verifiable.

Amplifying Privacy. In order to amplify the privacy guarantee of the above scheme, we consider a modified approach where we replace the random bits b and b' – which collide with probability one-half – with random strings of length ℓ that collide with $\frac{1}{2^\ell}$ probability. Specifically, consider a two-round protocol where the receiver's input is a random string $\mathbf{b}$ of length ℓ, while the sender also chooses a random string $\mathbf{b}'$ of length and "encrypts" some message m. Suppose that the protocol satisfies the following "extractability" property, namely, if $\mathbf{b}$ and $\mathbf{b}'$ are equal, then the receiver can extract the encrypted message; otherwise, m remains statistically hidden.

Now consider a modified version of our weakly-private two-round argument system where we replace the two-round OT with the above "string" variant. Note that with probability $1-2^\ell$, $\mathbf{b}$ and $\mathbf{b}'$ chosen by the prover and the verifier would be different, in which case, the α values would remain statistically hidden. This observation can, in fact, be turned into a formal proof for statistical witness indistiguishability.

The proof of computational soundness, however, now requires more work. Specifically, we now run the cheating prover for $\approx 2^\ell$ times, and estimate a $\mathbf{b}'_0$ that the cheating prover is most likely to output (with probability $\geq 1/2^\ell$). We then switch $\mathbf{b}$ to $\mathbf{b}'_0$. If the first round message of the receiver is secure against 2^ℓ-time adversaries, then the cheating prover would not notice the switch. We can now extract α values and derive a contradiction in a similar manner as before.

Two Round Public-Coin Statistical-Hiding Extractable Commitments. A two-round protocol that achieves statistical hiding property for the sender as well as extractability property of the aforementioned form was first formalized as a *statistical-hiding extractable commitment scheme* in the work of [KKS18]. Their construction, however, is private coin for the receiver. Below, we briefly recall their construction, and then discuss how it can be adapted to the public-coin setting.

- In the first round, the receiver samples a uniformly random string $\mathbf{b}$ of length ℓ. For each bit of the $\mathbf{b}$, the receiver sends a first round 1-out-of-2 OT message with the input bit specified by $\mathbf{b}$.

- The committer first samples a uniformly random string $\mathbf{b}'$ of length ℓ. To commit to a message m, the committer firstly uses the xor secret sharing to share m to ℓ shares. It then generates ℓ second round OT messages: for the i-th second round OT message, if the the i-th bit of $\mathbf{b}'$ is 0, then the committer puts the share in the first input slot, and puts a random value in the second slot. Otherwise, the committer puts the share in the second slot, and put a random value in the first slot.

From statistical sender-privacy of the underlying OT, the above construction achieves statistically hiding with probability $1 - 2^{\ell}$, even if the first round messages are maliciously generated.

Let us now explain the extractability property. For any committer, there exists a string $\mathbf{b}_0$ of length ℓ, such that the second string coincides with $\mathbf{b}_0$ with probability no less than $2^{-\ell}$. Therefore, we can switch the first round message of the commitment to hide $\mathbf{b}_0$. If we set ℓ to be sub-linear, and assume the first round message is secure against sub-exponential-time adversaries, then the committer would not notice the switching. Hence, when the two strings coincide, we can extract the committed message.

The aforementioned statistical-hiding extractable commitment scheme is a private coin scheme. To obtain a public-coin scheme, we rely on the fact that in many known statistical sender-private OT schemes, the first round message is pseudorandom. For example, in the recent construction of two-round statistical sender-private OT from LWE [BD18], the first round message is either statistical close to uniformly random, or is an LWE instance, which is computationally indistinguishable from the uniform distribution.

Putting it all together. Our final construction combines the above ideas to obtain a statistical Zap argument system:

- In the first round, the receiver simply sends the first round message of a two-round public-coin statistical-hiding extractable commitment scheme.
- Next, the prover samples a random string $\mathbf{b}'$ and computes a transcript of the sigma protocol in the same manner as before, except that it commits to α values within the second round messages of the public-coin statistical-hiding extractable commitment scheme.

We argue the statistical WI property by relying on the statistical-hiding property of the commitment scheme. The proof of soundness relies on the ideas discussed above. In order to base security on quasi-polynomial hardness assumptions, we set the parameter ℓ for the commitment scheme to be super-logarithmic rather than sub-linear. Given any cheating prover with inverse polynomial advantage, we run the cheating prover several times to estimate a string $\mathbf{b}_0$ of length ℓ such that the string chosen by the prover coincides with $\mathbf{b}_0$ with some inverse quasi-polynomial probability. This estimation takes quasi-polynomial time. Next, we switch the first round verifier message to one that is computed using $\mathbf{b}_0$. This switch is not noticeable to the prover since the first round message hides $\mathbf{b}_0$ even from adversaries that run in time 2^{ℓ}. This allows us to extract the α values and then invoke the correlation intractability of the CIH function as before. Note that we can construct the function f for CIH explicitly by using the receiver randomness for the first round message.

2.2 Three Round Statistical Receiver-Private OT

In this section, we describe our main ideas for constructing statistical receiver-private OT in three rounds in the plain model.

Prior work based on super-polynomial time assumptions. We start by briefly recalling the recent work of [KKS18] who investigated the problem of statistical receiver-private OT in three rounds. Since security w.r.t. black-box polynomial-time simulation is known to be impossible to achieve in three rounds [GK96], [KKS18] settled for the weaker goal of achieving security w.r.t. super-polynomial time simulation [Pas03]. To achieve their goal, [KKS18] implemented an OT reversal approach, starting from a two-round statistical sender-private OT to obtain a three-round statistical receiver-private OT based on super-polynomial-time hardness assumptions. In fact, the use of super-polynomial-time hardness assumptions seems somewhat inherent to their approach.

Motivated by our goal of basing security on standard polynomial-time hardness assumptions, we take a different approach, both in our security definition as well as techniques. On the definitional side, we consider distinguisher-dependent simulation security [DNRS99, JKKR17] for senders. On the technical side, we develop a general framework for three round statistical receiver-private OT via a new notion of *statistical hash commitment*. We elaborate on both of these aspects below.

Defining Security. In the setting of interactive proof systems, a well-studied security notion is weak zero-knowledge [DNRS99] which relaxes the standard notion of zero knowledge by reversing the order of quantifiers, namely, by allowing the simulator to depend upon the distinguisher. A recent work of [JKKR17] dubbed this idea as *distinguisher-dependent simulation* and studied it for proof systems and some other two-party functionalities. Following their approach, in this work, we formalize security for senders in three round OT via distinguisher-dependent simulation. Roughly speaking, this notion requires that for every malicious PPT receiver and PPT distinguisher, there must exist a PPT simulator that can simulate an indistinguishable view of the receiver.

Towards achieving distinguisher-dependent simulation security for senders, we first consider (computational) game-based security definition for senders. Interestingly, it is not immediately clear how to define game-based security for senders when we also require statistical receiver privacy. This is because in any protocol that achieves statistical receiver privacy, the protocol transcript does not fix the receiver message in an information-theoretic sense. As such, unlike the case of two-round computational receiver-private OT (where the receiver's input is information-theoretically fixed by the transcript), we cannot simply require indistinguishability of views generated using (say) sender inputs (m_b, m_{1-b}) and (m_b, m'_{1-b}), where b is presumably the input bit of the receiver.

This conundrum can be resolved by using an observation from [JKKR17]. In order to build proof systems with distinguisher-dependent simulation security, the work of [JKKR17] used the following natural property of two-round OT with computational privacy for senders and receivers – the distribution over receiver views generated using (say) sender inputs (m_0, m_1) must be indistinguishable from at least one of the following:

- Distribution over receiver views generated using sender inputs (m_0, m_0).
- Distribution over receiver views generated using sender inputs (m_1, m_1)

Intuitively, the first case corresponds to receiver input bit 0, while the second case corresponds to receiver input bit 1. It is not difficult to see that the above stated property is, in fact, meaningful even when the receiver's input is only fixed in a computational sense by the protocol transcript, which is the case in our setting. A recent work of [DGH+19] formulated a game-based security definition for senders that captures the above intuition, and we adopt it in this work. We also show that for our three round setting, game-based security for senders can be used to achieve distinguisher-dependent simulation security for senders.

So far, we have focused on formalizing security for senders. Formalizing security for receivers is easier; we consider game-based security that requires statistical/perfect indistinguishability of views generated with receiver inputs 0 and 1, against unbounded-time malicious senders.

In the remainder of this section, we describe our main ideas for constructing three-round OT with game-based security for senders and receivers.

A General Framework via Statistical Hash Commitment. We introduce a new notion of a statistical hash commitments (SHC) – a two-round statistically hiding commitment scheme where the decommitment verification simply involves an equality check with a hash output (computed w.r.t. a hashing algorithm associated with the scheme). We start by informally defining this notion and then discuss how it can be used to construct three-round OT with our desired security properties.

An SHC scheme is a two-round commitment scheme between a committer $\mathcal{C}$ and a receiver $\mathcal{R}$, that comes equipped with three additional algorithms – a key generation algorithm KGen, a commitment algorithm Com, and a hash algorithm H.

- In the first round, $\mathcal{R}$ samples a key pair $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}$ and sends pk to $\mathcal{C}$.
- In the second round, to commit a bit $b \in \{0,1\}$, the committer $\mathcal{C}$ executes $(c, \rho) \leftarrow \mathsf{Com}(\mathsf{pk}, b)$, and sends c to the receiver $\mathcal{R}$.
- In the opening phase, the committer $\mathcal{C}$ sends (b, ρ) to the receiver $\mathcal{R}$.
- The verification algorithm only involves an equality check: $\mathcal{R}$ computes the hash algorithm H using the private key k on input (c, b) and then matches the resulting value against ρ. If the check succeeds, then $\mathcal{R}$ accepts the opening, else it rejects.

We require an SHC scheme to satisfy the following two properties:

- *Computational Binding:* This property requires that no PPT malicious committer $\mathcal{C}$ can successfully compute a commitment c, and a opening ρ_0 and ρ_1 for *both* bits $b = 0$ and $b = 1$. Put differently, for an instance x and a second round message α, a PPT malicious committer cannot compute $\mathsf{H}(\mathsf{k}, c, b)$ for both $b = 0$ and $b = 1$.
- *Statistical (Perfect) Hiding:* This property requires that, every (possibly maliciously computed) public key pk, the commitment of 0 and 1 are statistically close.

Looking ahead, we use computational binding property of SHC to achieve computational game-based security for senders in our construction of three-round OT. The statistical (resp., perfect) hiding property, on the other hand, is used to achieve statistical (resp., perfect) game-based security for receivers.

From SHC to Three-Round OT. We next describe a generic transformation from an SHC scheme statistical/perfect receiver-private OT. In our protocol design, the OT sender plays the role of the receiver in SHC, while the OT receiver plays the role of the committer for SHC. In the discussion below, let b denote the input bit of the OT receiver and let (m_0, m_1) denote the input bits of the OT sender.

- In the first round, the sender samples a key pair $(\mathsf{pk}, \mathsf{k})$ using the key generation algorithm KGen for SHC, and sends pk to the sender.
- In the second round, it runs the commitment algorithm Com for SHC on input (pk, b) to compute a second round message c and an opening ρ, and sends c to the sender.
- In the last round, the sender samples two random strings (r_0, r_1) and then computes two "mask" bits z_0 and z_1, one each for its inputs m_0, m_1. The mask z_i (for $i \in \{0,1\}$) is computed as $\mathsf{hc}\big(\mathsf{H}(\mathsf{k}, c, i), r_i\big)$, where $\mathsf{hc}(\cdot,\cdot)$ is the Goldreich-Levin universal hardcore predicate [GL89].

To argue computational game-based security for senders, we crucially rely upon the strong soundness of SHC. In particular, the strong soundness of SHC, coupled with the security of the hardcore predicate ensures that at least one of the two mask bits z_i must be hidden from a malicious PPT receiver when the instance x is sampled from a hard distribution. Statistical (resp., perfect) security for receivers, on the other hand, follows from the statistical (resp., perfect) hiding property of the commitment.

We next discuss two different constructions of SHC.

Instantiating SHC from CDH. We first describe a construction of SHC that achieves *perfect* hiding property, based on CDH.

Let $\mathbf{M} = \begin{pmatrix} 1 & 0 \\ y & 1 \end{pmatrix}$, which must be full rank. Note that $g^{\mathbf{M}}$ can be computed using g^y.

- In the first round, the receiver $\mathcal{R}$ samples a random 2-by-1 column vector k as the secret key of the hash function, and sets the public key pk to be $\mathsf{pk} = (g^y, g^{\mathbf{M}\cdot\mathsf{k}})$. It then sends pk to the committer $\mathcal{C}$.

- The committer $\mathcal{C}$ (with input bit $b \in \{0,1\}$) samples a random 2-by-1 matrix $\boldsymbol{\alpha}$, and uses pk to compute $c = g^{\boldsymbol{\alpha}^T\cdot\mathbf{M}} \cdot g^{[0,b]}$. The committer sends c to the verifier, and then compute $\rho = g^{\boldsymbol{\alpha}^T\mathbf{M}\cdot\mathsf{k}}$

- The receiver $\mathcal{R}$ parse $c = g^{\mathbf{z}}$, and computes $\mathsf{H}(\mathsf{k}, c, b) = g^{(\mathbf{z}-[0,b])\cdot\mathsf{k}}$. If $\mathsf{H}(\mathsf{k}, c, b) = \rho$, then accept, otherwise reject.

We next informally argue the security of the above construction. Let us first consider computational binding property. Intuitively, for any prover who wants to compute two accepting last round messages ρ_0, ρ_1 for both $b = 0$ and $b = 1$, it must compute the inverse of $\mathbf{M}$, which requires that the prover knows the witness y. More formally, to prove the computational binding property, we build a PPT extractor that extracts y to derive a contradiction. Specifically, for any cheating committer who can output two accepting ρ_0, ρ_1 for $b = 0$ and $b = 1$, we can divide them to derive $g^{[0,1]\cdot\mathsf{k}}$. If we parse k as $\mathsf{k} = (s, t)$, then this implies that given $(g^y, g^{\mathbf{M}\dot{\mathsf{k}}}) = (g^y, g^{sy}, g^{sy+t})$, an efficient algorithm can compute $g^{[0,1]\cdot\mathsf{k}} = g^t$. We can then divide it from g^{sy+t} and derive g^{sy}. This gives us an efficient adversary for CDH.

To prove statistical hiding property, for any (potentially maliciously computed) pk, the commitment of bit $b \in \{0,1\}$ is $c = g^{\boldsymbol{\alpha}^T\cdot\mathbf{M}+[0,b]}$. Since the matrix $\mathbf{M}$ is full rank, and $\boldsymbol{\alpha}$ is uniformly random, we have that c is uniformly random. Hence, the commitment statistically hides b.

Instantiating SHC from Statistical Sender-Private 2-round OT. We next show a construction of SHC from any statistical sender-private 2-round OT protocol $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$, where OT_3 denotes the receiver output computation algorithm.

- In the first round, the receiver $\mathcal{R}$ samples a random string r of length ℓ. Then for each bit $r[i]$, it invokes OT_1 to generate a first round OT messsage $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, r[i])$. The public key pk is set to be the tuple of messages $\{\mathsf{ot}_{1,i}\}_{i\in[\ell]}$, while the private key k is set to be the tuple of private states $\{\mathsf{st}_i\}_{i\in[\ell]}$.

- The committer $\mathcal{C}$ receives pk, and its input is a bit b. It first samples a random string r' of length ℓ. For each position $i \in [\ell]$, it generates the second round OT messages $\mathsf{ot}_{2,i} = \mathsf{OT}_2(\mathsf{ot}_{1,i}, r'[i], r'[i] \oplus b)$. The commitment c is set to be the tuple of second round OT messages $\{\mathsf{ot}_{2,i}\}_{i\in[\ell]}$, and the opening $\rho = r'$.

- The verification process first computes $\mathsf{H}(\mathsf{k}, c, b)$ as follows: parse k as $\{\mathsf{st}_i\}_{i\in[\ell]}$, and the commitment c as $\{\mathsf{ot}_{2,i}\}_{i\in[\ell]}$. Then, compute $\rho_{0,i} \leftarrow \mathsf{OT}_3(\mathsf{ot}_{2,i}, \mathsf{st}_i)$, set $\rho_{1,i} = \rho_{0,i} \oplus r[i]$ for each $i \in [\ell]$, and set $\{\rho_{b,i}\}_{i\in[\ell]}$ to be the output of $\mathsf{H}(\mathsf{k}, c, b)$. If this output equals ρ, accept, otherwise, reject.

To show the completeness of this protocol, from the construction of the committer, we know that $\rho_{0,i} = r'[i] \oplus (r[i] \cdot b)$. From the computation of $\mathsf{H}(\mathsf{k}, c, b)$, we have that $\rho_{b,i} = \rho_{0,i} \oplus (r[i] \cdot b) = (r'[i] \oplus (r[i] \cdot b)) \oplus (r[i] \cdot b) = r'[i] = \rho$. The statistical hiding property follows from the statistical hiding property of the underlying OT. Finally, to show the construction is computational binding, our observation is that the construction of H always satisfies $\mathsf{H}(\mathsf{k}, c, 0) \oplus \mathsf{H}(\mathsf{k}, c, 1) = r$. Hence, any adversary breaking the computational binding property can also find $\rho_0 \oplus \rho_1 = \mathsf{H}(\mathsf{k}, c, 0) \oplus \mathsf{H}(\mathsf{k}, c, 1) = r$, given only the first round messages $\mathsf{ot}_{1,i}$. This breaks the computational receiver privacy of the OT.

3 Preliminaries

For any two (discrete) probability distributions P and Q, let $\mathsf{SD}(P, Q)$ denote *statistical distance* between P, Q. Let $\mathbb{Z}$ denote the set containing all integers. For any positive integer q, let $\mathbb{Z}_q$ denote the set $\mathbb{Z}/q\mathbb{Z}$. Let S be a discrete set, and let $\mathcal{U}(S)$ denote the uniform distribution over S. Throughout the paper, unless specified otherwise, we use λ to denote the security parameter.

3.1 Learning with Errors

We first recall the learning with errors (LWE) distribution.

Definition 1 (LWE distribution)**.** *For positive integer n and modulus q, and an error distribution χ over $\mathbb{Z}$, the LWE distribution $A_{\mathbf{s},\chi}$ is the following distribution. First sample a uniform random vector $\mathbf{a} \leftarrow \mathbb{Z}_q^n$, and an error $e \leftarrow \chi$, then output $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.*

Standard instantiations of LWE distribution usually choose χ to be discrete Gaussian distribution over $\mathbb{Z}$.

Definition 2 (Quasi-polynomial LWE Assumption)**.** *There exists a polynomial $n = n(\lambda)$ and a small real constant $c \in (0, 1/2)$ such that for any non-uniform probabilistic oracle adversary $\mathcal{D}^{(\cdot)}(\cdot)$ that runs in time $2^{O(\log^4 \lambda)}$, we have*

$$\mathsf{Adv}_\lambda(\mathcal{D}) = \left|\Pr\left[\mathcal{D}^{\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)}(1^\lambda) = 1\right] - \Pr\left[\mathbf{s} \leftarrow \mathbb{Z}_q^n : \mathcal{D}^{A_{\mathbf{s},\chi}}(1^\lambda) = 1\right]\right| < c$$

Where the adversary is given oracle access to the uniform distribution $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ or the LWE distribution $A_{\mathbf{s},\chi}$.

In the following Lemma 1, we show that quasi-polynomial LWE assumption implies that any adversary running in a slower quasi-polynomial time can only have inverse quasi-polynomial advantage.

Lemma 1. *Assuming quasi-polynomial hardness of LWE, for any non-uniform probabilistic adversary $\mathcal{D}$ that runs in time $2^{O(\log^2 \lambda)}$, we have*

$$\mathsf{Adv}_\lambda(\mathcal{D}) = \left|\Pr\left[\mathcal{D}^{\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)}(1^\lambda) = 1\right] - \Pr\left[\mathbf{s} \leftarrow \mathbb{Z}_q^n : \mathcal{D}^{A_{\mathbf{s},\chi}}(1^\lambda) = 1\right]\right| < 2^{-\Omega(\log^4 \lambda)}$$

Proof. We prove by contradiction. Suppose there exists an adversary $\mathcal{D}$ such that $\mathsf{Adv}_\lambda(\mathcal{D}) \geq 2^{-\log^4 \lambda}$ for infinitely many λ. Let $\epsilon = \mathsf{Adv}_\lambda(\mathcal{D})$. Then we construct the following adversary $\mathcal{D}'^{(\cdot)}(\cdot)$. The adversary $\mathcal{D}'$ is given access to an oracle $\mathcal{O}$, and is required to output a bit to tell if $\mathcal{O} = A_{\mathbf{s},\chi}$ or $\mathcal{O} = \mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$. The strategy of $\mathcal{D}'$ is described as follows.

Let $N_\lambda = 2^{100 \log^4 \lambda}$.

1. Execute $\mathcal{D}$ for N_λ times. In i-th execution, $i \in [N_\lambda]$, sample an $\mathbf{s}_i \leftarrow \mathbb{Z}_q^n$. Execute $\mathcal{D}^{\mathcal{O}}(1^\lambda)$ with fresh randomness. For each oracle query made by $\mathcal{D}$, forward the query to oracle $\mathcal{O}$, and then obtain a response $(\mathbf{a}, b)$. Let $b' = b + \langle \mathbf{a}, \mathbf{s}_i \rangle \in \mathbb{Z}_q$.[2] Forward $(\mathbf{a}, b')$ to $\mathcal{D}$. Let $S_{\mathcal{O}}$ be the number of executions where $\mathcal{D}$ outputs 1.

2. Execute $\mathcal{D}^{\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)}(1^\lambda)$ for N_λ times with fresh randomness for every execution. For each oracle query made by $\mathcal{D}$, sample an element uniform at random from $\mathbb{Z}_q^n \times \mathbb{Z}_q$, and forward the sample to $\mathcal{D}$. Let $S_{\mathcal{U}}$ be the number of executions where $\mathcal{D}$ outputs 1.

3. If $S_{\mathcal{O}} > S_{\mathcal{U}}$, output 1. If $S_{\mathcal{O}} < S_{\mathcal{U}}$, output 0. If $S_{\mathcal{O}} = S_{\mathcal{U}}$, output a random bit.

In the following, we assume $\Pr[\mathbf{s} \leftarrow \mathbb{Z}_q^n : \mathcal{D}^{A_{\mathbf{s},\chi}}(1^\lambda) = 1] = \Pr[\mathcal{D}^{\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)}(1^\lambda) = 1] + \epsilon$. The proof for the other case follows in the same manner, and is omitted.

When $\mathcal{O} = \mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$, $S_{\mathcal{O}}$ and $S_{\mathcal{U}}$ are subjected to two independent and identical distributions. Thus, $\mathcal{D}'^{\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)}(1^\lambda)$ outputs a random bit. We have that $\Pr[\mathcal{D}'^{\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)}(1^\lambda) = 1] = 1/2$.

When $\mathcal{O} = A_{\mathbf{s},\chi}$, denote $\mu_O = E[S_{\mathcal{O}}]$, $\mu_U = E[S_{\mathcal{U}}]$. Now we lower bound the probability

$$\begin{aligned}\Pr[\mathcal{D}'^{A_{\mathbf{s},\chi}}(1^\lambda) = 1] = 1 - \Pr[\mathcal{D}'^{A_{\mathbf{s},\chi}}(1^\lambda) = 0] &\geq 1 - \Pr[S_{\mathcal{O}} \leq S_{\mathcal{U}}] \\ &\geq 1 - \left(\Pr\left[S_{\mathcal{O}} \leq \frac{\mu_O + \mu_U}{2}\right] + \Pr\left[S_{\mathcal{U}} \geq \frac{\mu_O + \mu_U}{2}\right]\right)\end{aligned}$$

The first line comes from the fact that $\mathcal{D}'$ outputs 0 only when $S_{\mathcal{O}} < S_{\mathcal{U}}$ or $S_{\mathcal{O}} = S_{\mathcal{U}}$. The second line follows from a union bound, since $S_{\mathcal{O}} \leq S_{\mathcal{U}}$ implies $S_{\mathcal{O}} \leq \frac{\mu_P + \mu_U}{2}$ or $S_{\mathcal{U}} \geq \frac{\mu_P + \mu_U}{2}$.

From Chernoff bound, we have

$$\Pr\left[S_{\mathcal{O}} \leq \frac{\mu_O + \mu_U}{2}\right] \leq \exp\left(-\frac{1}{2}\left(\frac{\mu_O - \mu_U}{2\mu_O}\right)^2 \mu_O\right) \leq \exp\left(-\frac{1}{8}\epsilon^2 N\right)$$

$$\Pr\left[S_{\mathcal{U}} \geq \frac{\mu_O + \mu_U}{2}\right] \leq \exp\left(-\frac{\left(\frac{\mu_O - \mu_U}{2\mu_U}\right)^2}{2 + \frac{\mu_O - \mu_U}{2\mu_U}} \mu_U\right) \leq \exp\left(-\left(\frac{1}{2}\epsilon^2 + O(\epsilon^3)\right) N\right)$$

Hence, $\Pr[\mathcal{D}'^{A_{\mathbf{s},\chi}}(1^\lambda) = 1] \geq 1 - \exp(-\Omega(\epsilon^2 N))$. Thus, we have $\mathsf{Adv}_\lambda(\mathcal{D}') \geq 1/2 - \exp(-\Omega(\epsilon^2 N)) = 1/2 - \mathsf{neg}(\lambda)$. Note that $\mathcal{D}'$ runs in time $2^{O(\log^4 \lambda)}$. We reach a contradiction with quasi-polynomial LWE assumption. □

3.2 Computational Diffie-Hellman Assumption

Definition 3. *Let G be a cyclic group of order q generated by g, where each element of G can represented in a polynomial $n = n(\lambda)$ number of bits. The CDH assumption states that for any non-uniform PPT adverrsary $\mathcal{A}$, there exists an negligible function $\nu(\lambda)$ such that*

$$\Pr[x \leftarrow \mathbb{Z}_q, y \leftarrow \mathbb{Z}_q, z \leftarrow \mathcal{A}(1^\lambda, g^x, g^y) : z = g^{xy}] < \nu(\lambda)$$

[2] Here, we use the worst-case to average-case reduction for LWE [Reg05].

3.3 Goldreich-Levin Hardcore Predicate

Definition 4. *Let f be an one-way function from $\{0,1\}^n \to \{0,1\}^m$, where $n = n(\lambda)$ and $m = m(\lambda)$ are polynomials of λ. The Goldreich-Levin hardcore predicate* hc *is defined as* $\mathsf{hc}(x,r) = \langle x, r\rangle_2$, *where $x, r \in \{0,1\}^n$, and $\langle \cdot, \cdot \rangle_2$ is the inner product function modulo 2.*

Theorem 3 (Goldreich-Levin Theorem [GL89], modified)**.** *If there exists an PPT adversary $\mathcal{A}$ such that*

$$\Pr[x \leftarrow \{0,1\}^n, r \leftarrow \{0,1\}^n, b \leftarrow \mathcal{A}(1^\lambda, (f(x), r)) : b = \mathsf{hc}(x,r)] > 1/2 + \epsilon(\lambda)$$

where $\epsilon(\lambda)$ is an non-negligible function of λ, then there exists a PPT inverter $\mathcal{A}'$ s.t.

$$\Pr[x \leftarrow \{0,1\}^n, x' \leftarrow \mathcal{A}'(1^\lambda, f(x)) : x' = x] > \epsilon'(\lambda)$$

where $\epsilon'(\lambda)$ is also an non-negligible function λ.

3.4 Statistical Zap Arguments

Zaps [DN00] are two-round witness indistinguishable proof systems with a public-coin verifier message. Below, we define statistical Zap arguments, i.e., Zaps that achieve statistical WI property and computational soundness.

Let $\mathcal{P}$ denote the prover and $\mathcal{V}$ denote the verifier. We use $\mathsf{Trans}(\mathcal{P}(1^\lambda, x, \omega) \leftrightarrow \mathcal{V}(1^\lambda, x))$ to denote the transcript of an execution between $\mathcal{P}$ and $\mathcal{V}$, where $\mathcal{P}$ and $\mathcal{V}$ both have input a statement x and P also has a witness ω for x.

Definition 5. *Let L be a language in* NP*. We say that a two round protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ with a public-coin verifier message is a statistical Zap argument for L if it satisfies the following properties:*

Completeness *For every $x \in L$, and witness ω for x, we have that*

$$\Pr\left[\mathsf{Trans}(\mathcal{P}(1^\lambda, x, \omega) \leftrightarrow \mathcal{V}(1^\lambda, x)) \textit{ is accepted by } \mathcal{V}\right] = 1$$

Computational Soundness *For any non-uniform probabilistic polynomial time (cheating) prover $\mathcal{P}^*$, there exists a negligible function $\nu(\cdot)$ such that for any $x \notin L$, we have that*

$$\Pr\left[\mathsf{Trans}(\mathcal{P}^*(1^\lambda, x) \leftrightarrow \mathcal{V}(1^\lambda, x)) \textit{ is accepted by } \mathcal{V}\right] < \nu(\lambda)$$

Statistical Witness Indistinguishability *For any (unbounded cheating) verifier $\mathcal{V}^*$, there exists a negligible function $\nu(\cdot)$ such that for every $x \in L$, and witnesses ω_1, ω_2 for x, we have that*

$$\mathsf{SD}\left(\mathsf{Trans}(\mathcal{P}(1^\lambda, x, \omega_1) \leftrightarrow \mathcal{V}^*(1^\lambda, x)), \mathsf{Trans}(\mathcal{P}(1^\lambda, x, \omega_2) \leftrightarrow \mathcal{V}^*(1^\lambda, x))\right) < \nu(\lambda)$$

3.5 Statistical Sender-Private Oblivious Transfer

Definition 6. *A statistical sender-private oblivious transfer (OT) is a tuple of algorithms* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$*:*

$\mathsf{OT}_1(1^\lambda, b)$**:** *On input security parameter λ, a bit $b \in \{0,1\}$,* OT_1 *outputs the first round message* ot_1 *and a state* st.

$\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, m_0, m_1)$**:** *On input security parameter λ, a first round message ot_1, two bits $m_0, m_1 \in \{0,1\}$, OT_2 outputs the second round message ot_2.*

$\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st})$**:** *On input security parameter λ, the second round message ot_2, and the state generated by OT_1, OT_3 outputs a message m.*

We require the following properties:

Correctness *For any $b, m_0, m_1 \in \{0,1\}$,*

$$\Pr\left[\substack{(\mathsf{ot}_1,\mathsf{st})\leftarrow\mathsf{OT}_1(1^\lambda,b),\mathsf{ot}_2\leftarrow\mathsf{OT}_2(1^\lambda,\mathsf{ot}_1,m_0,m_1),\\ m\leftarrow\mathsf{OT}_3(1^\lambda,\mathsf{ot}_2,\mathsf{st})} : m = m_b\right] = 1$$

Statistical Sender Privacy *There exists a negligible function $\nu(\lambda)$ and an deterministic exponential time extractor OTExt such that for any (potential maliciously generated) ot_1, $\mathsf{OTExt}(1^\lambda, \mathsf{ot}_1)$ outputs a bit $b \in \{0,1\}$. Then for any $m_0, m_1 \in \{0,1\}$, we have*

$$\mathsf{SD}\left(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, m_0, m_1), \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, m_b, m_b)\right) < \nu(\lambda)$$

Quasi-polynomial Pseudorandom Receiver's Message *For any $b \in \{0,1\}$, let ot_1 be the first round message generated by $\mathsf{OT}_1(1^\lambda, b)$. For any non-uniform probabilistic adversary $\mathcal{D}$ that runs in time $2^{O(\log^2 \lambda)}$, we have*

$$\mathsf{Adv}_\lambda(\mathcal{D}) = \left| \Pr\left[\mathcal{D}(1^\lambda, \mathsf{ot}_1) = 1\right] - \Pr\left[u \leftarrow \{0,1\}^{|\mathsf{ot}_1|} : \mathcal{D}(1^\lambda, u) = 1\right] \right| < 2^{-\Omega(\log^4 \lambda)}$$

Lemma 2. *Assuming quasi-polynomial hardness of LWE, there exists a statistical sender private oblivious transfer scheme.*

A statistical sender-private OT scheme from LWE was recently constructed by [BD18]. Their construction satisfies correctness and statistical sender-privacy. Further, the receiver's message in their scheme is pseudorandom, assuming LWE. We observe that assuming quasi-polynomial LWE and using Lemma 1, their scheme also satisfies quasi-polynomially pseudorandom receiver's message property.

3.6 Correlation Intractable Hash Function

The following definition is taken verbatim from [PS19].

Definition 7 (Searchable Relation [PS19])**.** *We say that a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ is searchable in* size S *if there exists a function $f : \mathcal{X} \to \mathcal{Y}$ that is implementable as a* Boolean *circuit of size S, such that if $(x, y) \in R$ then $y = f(x)$.*

Correlation intractable hash function is a family of keyed hash functions satisfying the following property: for any searchable relation R, it is hard for a computationally unbounded adversary to find an element x such that $(x, f(x)) \in R$.

Definition 8 (Correlation Intractable Hash Function, slightly modified from [PS19])**.** Correlation Intractable Hash Function *(CIH) is a triple of algorithms* $(\mathsf{KGen}, \mathsf{FakeGen}, \mathsf{H}_{(\cdot)}(\cdot))$*, with the following properties:*

Let $s = s(\lambda), \ell = \ell(\lambda), d = d(\lambda)$ be $\mathsf{poly}(\lambda)$-bounded functions. Let $\{\mathcal{R}_{\lambda,s,\ell,d}\}_\lambda$ be a family of searchable relations, where each relation $R \in \mathcal{R}_{\lambda,s,\ell,d}$ is searchable by a circuit of size $s(\lambda)$, output length $\ell(\lambda)$ and depth $d(\lambda)$.

Statistical Correlation Intractable *There exists a negligible function $\nu(\cdot)$ such that, for any relation $R \in \mathcal{R}_{\lambda,s,\ell,d}$, and circuit C_λ that searches for a witness for R, we have*

$$\Pr\left[k \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|C_\lambda|}, C_\lambda) : \exists x \ s.t. \ (x, \mathsf{H}_k(x)) \in R\right] < \nu(\lambda)$$

Quasi-polynomial Pseudorandom Fake Key *For any circuit C_λ with size s, output length ℓ, and depth d, $\mathsf{KGen}(1^\lambda, 1^{|C_\lambda|})$ outputs an uniform random string. Furthermore, for any non-uniform adversary $\mathcal{D}$ that runs in time $2^{O(\log^2 \lambda)}$, we have*

$$\left|\Pr\left[\mathcal{D}(1^\lambda, 1^{|C_\lambda|}, \mathsf{KGen}(1^\lambda, 1^{|C_\lambda|})) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, 1^{|C_\lambda|}, \mathsf{FakeGen}(1^\lambda, 1^{|C_\lambda|}, C_\lambda)) = 1\right]\right| \leq 2^{-\Omega(\log^4 \lambda)}$$

Theorem 4. *Assuming quasi-polynomial hardness of LWE, there exists a construction of correlation intractable hash function with quasi-polynomial pseudorandom fake key.*

The construction of such a function is given in [PS19, CCH+19]. Specifically, we use the construction of [PS19], which satisfies *statistical correlation intractability.* Moreover, the $\mathsf{FakeGen}$ algorithm in their construction simply consists of some ciphertexts that are pseudorandom assuming LWE. Thus, if we assume quasi-polynomial hardness of LWE, their construction satisfies quasi-polynomial pseudorandom fake key property.

For our application, we require a slightly stronger property than statistical correlation intractability as defined above. Specifically, we require that the distinguishing probability in statistical correlation intractability is $2^{-\lambda}$ for a special class of relations.

We show in Lemma 3 that by using parallel repetition, we can construct a CIH with the above property from any CIH.

Lemma 3 (Amplification of Statistical Correlation Intractability)**.** *There exists a correlation intractable hash function* $(\mathsf{KGen}, \mathsf{FakeGen}, \mathsf{H}_{(\cdot)}(\cdot))$ *such that the following additional property holds.*

$2^{-\lambda}$-Statistical Correlation Intractability *Let $\{C_\lambda\}_\lambda$ be a family of* Boolean *circuits, where C_λ has polynomial size $s(\lambda)$, polynomial depth $d(\lambda)$, and outputs a single bit. There exists a polynomial $\ell = \ell(\lambda)$ such that the following holds. Let $\overrightarrow{C_{\lambda,\ell}}$ be the circuit $\overrightarrow{C_\lambda}(c_1, c_2, \ldots, c_\ell) = (C_\lambda(c_1), C_\lambda(c_2), \ldots, C_\lambda(c_\ell))$, then for large enough λ,*

$$\Pr\left[k \leftarrow \mathsf{FakeGen}\left(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell}}|}, \overrightarrow{C_{\lambda,\ell}}\right) : \exists x \ s.t. \ \mathsf{H}_k(x) = \overrightarrow{C_{\lambda,\ell}}(x)\right] < 2^{-\lambda}$$

The CIH in [PS19] already satisfies the above property. In the following, we describe a generic transformation from any CIH to one that achieves the above property.

Proof. Let C_{in} be the length of input to C_λ. We prove this corollary from any CIH (KGen', $\mathsf{FakeGen}', \mathsf{H}'_{(\cdot)}(\cdot)$), where H' is a hash function family $\{0,1\}^{C_{in}\cdot\ell'} \to \{0,1\}^{\ell'}$. Denote $R_{\overrightarrow{C_{\lambda,\ell'}}} = \left\{(x, \overrightarrow{C_{\lambda,\ell'}}(x))\right\}$. We construct the following new CIH.

Parameters Set $\ell(\lambda) = \ell'(\lambda) \cdot \lambda$.

$\mathsf{KGen}(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell}}|}, \overrightarrow{C_{\lambda,\ell}})$: For each $i \in [\lambda]$, execute $k_i \leftarrow \mathsf{KGen}'(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell'}}|}, \overrightarrow{C_{\lambda,\ell'}})$ with fresh randomness. Output $k = (k_i)_{i\in[\lambda]}$.

$\mathsf{FakeGen}(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell}}|}, \overrightarrow{C_{\lambda,\ell}})$: For each $i \in [\lambda]$, execute $k_i \leftarrow \mathsf{FakeGen}'(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell'}}|}, \overrightarrow{C_{\lambda,\ell'}})$ with fresh randomness. Output $k = (k_i)_{i\in[\lambda]}$.

$\mathsf{H}_k(c_1, c_2, \ldots, c_\ell)$: For each $i \in [\lambda]$, execute $\mathbf{b}_i = \mathsf{H}_k(c_{\ell'(i-1)+1}, c_{\ell'(i-1)+2}, \ldots, c_{\ell' i})$, output $\mathbf{b} = (\mathbf{b}_i)_{i \in [\lambda]}$.

We now prove that the above construction satisfies $2^{-\lambda}$-statistical correlation intractability. For large enough λ, $\nu(\lambda) < 1/2$. Hence we have

$$\begin{aligned} &\Pr\left[k \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell}}|}, \overrightarrow{C_{\lambda,\ell}}) : \exists x = (x_i)_{i \in [\ell]} \text{ s.t. } \mathsf{H}_k(x) = \overrightarrow{C_{\lambda,\ell}}(x)\right] \\ = &\Pr\left[\forall i \in [\lambda], k_i \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|\overrightarrow{C_{\lambda,\ell'}}|}, \overrightarrow{C_{\lambda,\ell'}}) : \exists x_i, x_i \in R_{\overrightarrow{C_{\lambda,\ell'}}}\right] \leq (\nu(\lambda))^\lambda < 2^{-\lambda} \end{aligned}$$

The second line follows from the fact that k_i are generated independently. □

4 Statistical Zap Arguments

4.1 Public Coin Statistical-Hiding Extractable Commitments

In this section, we start by defining and constructing a key building block in our construction of statistical Zaps, namely, a statistical-hiding extractable commitment scheme. The notion and its construction are adapted from [KKS18], with some slight modifications to fit in our application. The main difference between our definition and that of [KKS18] is that we require the first round message to be public coin as opposed to private-coin.

Our syntax departs from the classical definition of commitment schemes. We consider a tuple of four algorithms $(\mathsf{Com}_1, \mathsf{FakeCom}_1, \mathsf{Com}_2, \mathsf{Dec})$, where Com_1 corresponds to the honest receiver's algorithm that simply outputs a uniformly random string. Com_2 corresponds to the committer's algorithm that takes as input a message m as well as a random string $\mathbf{b}'$ of length μ and outputs a commitment string. We require two additional algorithms: (1) $\mathsf{FakeCom}_1$ that takes a binary string $\mathbf{b}$ of length μ as input and produces a first round message that "hides" the string $\mathbf{b}$, and (2) Dec that takes as input a transcript generated using $\mathsf{FakeCom}_1$ and Com_2 and outputs the committed message if the strings $\mathbf{b}$ and $\mathbf{b}'$ used for computing the transcript are equal.

Let $\mathcal{C}$, $\mathcal{R}$ denote the committer and the receiver, respectively. We now proceed to give a formal definition.

Definition 9. *A public coin statistical-hiding extractable commitment is a tuple* $(\mathsf{Com}_1, \mathsf{FakeCom}_1, \mathsf{Com}_2, \mathsf{Dec})$. *The commit phase and open phase are defined as follows.*

Commitment Phase

Round 1 *On input parameters* $(1^\lambda, 1^\mu)$, $\mathcal{R}$ *executes* Com_1 *to sample a uniform random string* com_1. $\mathcal{R}$ *sends* com_1 *to* $\mathcal{C}$.

Round 2 *On input* $(1^\lambda, m)$, $\mathcal{C}$ *chooses* $\mathbf{b}' \leftarrow \{0,1\}^\mu$ *uniformly at random.*

Computes $\mathsf{com}_2 \leftarrow \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m; r)$ *with randomness* r.

$\mathcal{C}$ *sends* $(\mathbf{b}', \mathsf{com}_2)$ *to* $\mathcal{R}$.

Opening Phase

$\mathcal{C}$ *sends the message and the randomness* (m, r) *to* $\mathcal{R}$.
$\mathcal{R}$ *checks if* $\mathsf{com}_2 = \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m; r)$.

We require the following properties of the commitment scheme.

Statistical Hiding *There exists a negligible function $\nu(\cdot)$, a deterministic exponential time algorithm* ComExt*, and a randomized simulator* Sim*, such that for any fixed (potentially maliciously generated)* com_1*,* $\mathsf{ComExt}(1^\lambda, 1^\mu, \mathsf{com}_1)$ *outputs* $\mathbf{b} \in \{0,1\}^\mu$*, and for any* $\mathbf{b}' \neq \mathbf{b}$*, and* $m \in \{0,1\}$*, we have*

$$\mathsf{SD}\left(\mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m), \mathsf{Sim}(1^\lambda, 1^\mu, \mathsf{com}_1)\right) < \mu \cdot \nu(\lambda) \tag{1}$$

Quasi-polynomial Pseudorandom Receiver's Message *For any* $\mathbf{b} \in \{0,1\}^\mu$*,* $\mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b})$ *and a uniform random string outputted by* $\mathsf{Com}(1^\lambda, 1^\mu)$ *are quasi-polynomially indistinguishable. Specifically, for any non-uniform adversary* $\mathcal{D}$ *that runs in time* $2^{O(\log^2 \lambda)}$*, we have*

$$\left|\Pr[\mathcal{D}(1^\lambda, 1^\mu, \mathsf{Com}_1(1^\lambda, 1^\mu)) = 1] - \Pr[\mathcal{D}(1^\lambda, 1^\mu, \mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b})) = 1]\right| \leq \mu \cdot 2^{-\Omega(\log^4 \lambda)}$$

Extractable $\mathsf{FakeCom}_1$ *and* Dec *satisfy the following property. For any* $\mathbf{b} \in \{0,1\}^\mu$*, we have*

$$\Pr\left[\substack{(\mathsf{com}_1, \mathsf{st}) \leftarrow \mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b}),\\ \mathsf{com}_2 \leftarrow \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}, m)} : \mathsf{Dec}(1^\lambda, 1^\mu, \mathsf{st}, \mathsf{com}_2) = m\right] = 1$$

Lemma 4. *Assuming quasi-polynomial hardness of LWE, there exists a public coin statistical-hiding extractable commitment scheme.*

We construct a public coin statistical hiding extractable commitment by slightly modifying the commitment scheme of [KKS18]. Their construction already satisfies extractability and statistical hiding properties. However, their construction, as originally described, is private coin. We note that the receiver's message in their scheme simply consists of multiple receiver messages of a statistical sender-private OT scheme. Then, by instantiating their construction with an OT scheme that satisfies quasi-polynomial pseudorandom receiver's message property (see Section 3.5), their scheme can be easily adapted to obtain a *public coin* statistical-hiding extractable commitment. Specifically, in the modified construction, the honest receiver's algorithm $\mathsf{Com}(1^\lambda, 1^\mu)$ simply computes a uniform random string, while $\mathsf{FakeCom}_1$ corresponds to the receiver algorithm in the construction of [KKS18].

Construction. For completeness, here we describe the construction adapted from [KKS18].

$\mathsf{Com}_1(1^\lambda, 1^\mu)$ **:** Output a uniform random string $\mathsf{com}_1 \leftarrow \{0,1\}^{|\mathsf{com}_1|}$.

$\mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b})$ **:** Parse $\mathbf{b} = (b_1, b_2, \ldots, b_\mu)$. For each $i \in [\mu]$, execute $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, b_i)$. Output $\mathsf{com}_1 = (\mathsf{ot}_{1,i})_{i \in [\mu]}$ and $\mathsf{st} = (\mathsf{st}_i)_{i \in [\mu]}$.

$\mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m)$ **:** Parse $\mathbf{b}' = (b_1', b_2', \ldots, b_\mu')$, and $\mathsf{com}_1 = (\mathsf{ot}_{1,i})_{i \in [\mu]}$. Sample uniform random $m_1, m_2, \ldots, m_\mu \in \{0,1\}$ such that $\bigoplus_{i \in [\mu]} m_i = m$. For each $i \in [\mu]$, let $m_{b_i',i} = m_i$, and sample $m_{1-b_i',i} \leftarrow \{0,1\}$. Execute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(1^\lambda, 1^\mu, m_{0,i}, m_{1,i})$. Output $\mathsf{com}_2 := (\mathsf{ot}_{2,i})_{i \in [\mu]}$.

$\mathsf{Dec}(1^\lambda, 1^\mu, \mathsf{st}, \mathsf{com}_2)$ **:** Parse $\mathsf{st} = (\mathsf{st}_i)_{i \in [\mu]}$, and $\mathsf{com}_2 = (\mathsf{ot}_{2,i})_{i \in [\mu]}$. For each $i \in [\mu]$, execute $m_i' \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_{2,i}, \mathsf{st}_i)$. Let $m' = \bigoplus_{i \in [\mu]} m_i'$. Output m'.

This completes the description of the scheme.

Theorem 5. *The above construction satisfies statistical-hiding, quasi-polynomial pseudorandom receiver's message property and extractability.*

Proof. We now argue each of the three properties separately.

Statistical Hiding We construct the following extracting algorithm $\mathsf{ComExt}(1^\lambda, 1^\mu, \mathsf{com}_1 = (\mathsf{ot}_{1,i})_{i\in[\mu]})$. For each $i \in [\mu]$, execute $b_i = \mathsf{OTExt}(1^\lambda, \mathsf{ot}_{1,i})$. Output $\mathbf{b} = (b_i)_{i\in[\mu]}$.

Let $\mathbf{b} = \mathsf{ComExt}(1^\lambda, 1^\mu, \mathsf{com}_1)$, then for any $\mathbf{b}' \neq \mathbf{b}$, consider the following hybrids.

$\mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m)$ **:** Sample $(m_i)_{i\in[\mu]}$ uniformly at random such that $\bigoplus_{i\in[\mu]} m_i = m$. For each $i \in [\mu]$, set $m_{i,b'_i} = m_i$, and $m_{i,1-b'_i} \leftarrow \{0,1\}$. Output $(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_{1,i}, m_{i,0}, m_{i,1}))_{i\in[\mu]}$.

$\mathsf{Hybrid}(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m)$ **:** Sample $(m_i)_{i\in[\mu]}$ uniformly at random such that $\bigoplus_{i\in[\mu]} m_i = m$. For each $i \in [\mu]$, set $m_{i,b'_i} = m_i$, and $m_{i,1-b'_i} \leftarrow \{0,1\}$. Output $(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_{1,i}, \underline{m_{i,b_i}, m_{i,b_i}}))_{i\in[\mu]}$.

$\mathsf{Sim}(1^\lambda, 1^\mu, \mathsf{com}_1)$ **:** $\underline{\text{Sample } m_1, m_2, \ldots, m_\mu \leftarrow \{0,1\}}$. Output $(\mathsf{OT}_2(1^\lambda, \mathsf{ot}_{1,i}, m_i, m_i))_{i\in[\mu]}$.

From the statistical-hiding property of underlying OT scheme, it follows that Com_2 and Hybrid are statistically close. Specifically, there exists a negligible function $\nu(\cdot)$ such that:

$$\mathsf{SD}\left(\mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m), \mathsf{Hybrid}(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m)\right) < \mu \cdot \nu(\lambda)$$

Next, we prove that Hybrid and Sim are identical distributions. Denote $\mathcal{I} = \{i^* \in [\mu] | b_{i^*} \neq b'_{i^*}\}$. Since $\mathbf{b} \neq \mathbf{b}'$, we have $\mathcal{I} \neq \phi$. Hence, the joint distribution $(m_{i,b_i})_{i\in[\mu]\setminus\mathcal{I}}$ is uniformly random. Since $b_{i^*} \neq b'_{i^*}$ for all $i^* \in \mathcal{I}$, $(m_{i^*,b_{i^*}})$ is sampled uniformly at random for all $i^* \in \mathcal{I}$. Hence, $(m_{i,b_i})_{i\in[\mu]}$ is uniformly random. Hence, $\mathsf{Hybrid}(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', m)$ and $\mathsf{Sim}(1^\lambda, 1^\mu, \mathsf{com}_1)$ are identical distributions.

The statistical hiding property of the construction now follows by combining the above claims.

Quasi-polynomial Pseudorandom Receiver's Message This property directly follows from the quasi-polynomial pseudorandom receiver message property the OT scheme.

Extractable This property directly follows from the correctness of the OT scheme.

□

4.2 Our Construction

In this section, we describe our construction of a statistical Zap argument system for Graph Hamiltonicity, which is an NP-Complete problem.

Notation. We describe some notation that is used in our construction. Let L_{HAM} denote the Graph Hamiltonicity language over graphs $G = (V, E)$ of n vertices, where V denotes the set of vertices and E denotes the set of edges in G. We slightly abuse notation and use G to denote its adjacency matrix $G = (G_i[s,t])_{s,t\in[n]}$.

Let $(\mathsf{Com}_1, \mathsf{FakeCom}_1, \mathsf{Com}_2, \mathsf{Dec})$ be a public coin statistical-hiding extractable commitment scheme (Definition 9). We set the parameter μ of the commitment scheme as $\Theta(\log^2 \lambda)$. Let $(\mathsf{KGen}, \mathsf{FakeGen}, \mathsf{H}_{(\cdot)}(\cdot))$ be a family of CIH (Definition 8). We choose the polynomial $\ell = \ell(\lambda)$ in Lemma 3 such that the CIH is $2^{-\lambda}$-statistical correlation intractable.

Circuit C_{st}. Let C_{st} denote the following Boolean circuit.

Input: a $n \times n$ matrix $c = (c_{s,t})_{s,t\in[n]}$.

Output: a boolean value.

1. For any $s, t \in [n]$, execute $G[s,t] = \mathsf{Dec}(1^\lambda, 1^\mu, \mathsf{st}, c_{s,t})$.
2. If $G = (G_i[s,t])_{s,t\in[n]}$ is a cycle graph, then output 0. Otherwise output 1.

For ease of exposition, we extend the notation C_{st} to a series of matrices $(c_1, c_2, \ldots, c_\ell)$. Specifically, $C_{\mathsf{st}}(c_1, c_2, \ldots, c_\ell)$ is defined as $(C_{\mathsf{st}}(c_1), C_{\mathsf{st}}(c_2), \ldots, C_{\mathsf{st}}(c_\ell))$.

Construction. The verifier $\mathcal{V}$ and prover $\mathcal{P}$ are both given input the security parameter λ and a graph $G = (V, E)$ of n vertices. The prover is additionally given as input a witness ω for G.

Round 1 Verifier $\mathcal{V}$ computes and sends uniform random strings $(\mathsf{com}_1 \leftarrow \mathsf{Com}_1(1^\lambda, 1^\mu), k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|})$, where C_{st} takes ℓ separate $n \times n$ matrices as input, and outputs ℓ bits.

Round 2 Prover $\mathcal{P}$ does the following:

1. Choose a random $\mathbf{b}' \leftarrow \{0,1\}^\mu$.
2. Compute ℓ first round messages of Blum's sigma protocol for Graph Hamiltonicity. Specifically, for every $i \in [\ell]$, first sample a random cycle graph $G_i = (G_i[s,t])_{s,t\in[n]}$. Next, for each $s, t \in [n]$, compute $\mathsf{c}_i[s,t] \leftarrow \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', G_i[s,t]; r_i^{(s,t)})$ using randomness $r_i^{(s,t)}$. Finally let $\mathsf{c}_i = (\mathsf{c}_i[s,t])_{s,t\in[n]}$.
3. Compute $(b_1, b_2, \ldots, b_\ell) = \mathsf{H}_k(\mathsf{c}_1, \ldots, \mathsf{c}_\ell)$.
4. For every $i \in [\ell]$, compute the answer to challenge b_i in Blum's sigma protocol. Specifically, if $b_i = 0$, then set $z_i = (G_i, (r_i^{(s,t)})_{s,t\in[n]})$. Else, if $b_i = 1$, then compute a one-to-one map $\phi : G \to G_i$ such that $\phi(w)$ is the cycle G_i, and set $z_i = (\phi, (r_i^{(s,t)})_{(s,t)=\phi(e), e\notin E})$.
5. Send $\Pi = (\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ to the verifier.

Verification Upon receiving the proof $\Pi = (\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$, the verifier first computes $(b_1, b_2, \cdots, b_\ell) = \mathsf{H}_k(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell)$, and then verifies each copy (c_i, b_i, z_i) of the proof as in Blum's protocol.

Specifically, if $b_i = 0$, then parse $z_i = (G_i, (r_i^{(s,t)})_{s,t\in[n]})$ and check if $\mathsf{c}_i = (\mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', G_i[s,t]; r_i^{(s,t)})_{s,t\in[n]}$ and G_i is a cycle graph. Otherwise if $b_i = 1$, then parse $z_i = (\phi, (r_i^{(s,t)})_{(s,t)=\phi(e), e\notin E})$ and check if ϕ is a one-to-one map, and for each $e \notin E$, and $(s,t) = \phi(e)$, check if $\mathsf{c}_i[s,t] = \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', 0; r_i^{(s,t)})$. If all of the checks succeed, then accept the proof, otherwise reject.

This completes the description of our construction.

Theorem 6 (Completeness)**.** *The construction in Section 4.2 satisfies completeness.*

In our construction, both the prover and the verifier compute the challenges as $(b_1, b_2, \ldots, b_\ell) = \mathsf{H}_k(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell)$. Hence, to prove that the verification succeeds, it suffices to prove that for each $i \in [\ell]$, z_i is a valid answer to c_i for the challenge b_i. In a nutshell, this follows from the completeness of Blum's sigma protocol.

More specifically, if $b_i = 0$, then in step 2, $\mathcal{P}$ computes $\mathsf{c}_i = (\mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', G_i[s,t]; r_i^{(s,t)}))_{s,t\in[n]}$ honestly with a random cycle graph G_i. Therefore, the verification in this case succeeds. Otherwise if $b_i = 1$, we need to show that $\mathsf{c}_i[s,t] = \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', 0; r_i^{(s,t)})$ for every $e \notin E$ and $(s,t) = \phi(e)$. It suffices to show that $G_i[s,t] = 0$ for such (s,t). Note that if $e \notin E$, then $\phi(e) \notin \phi(G)$, since ϕ is a one-to-one map. Hence, if $(s,t) = \phi(e)$, then $G_i[s,t] = 0$. This completes the proof.

Theorem 7 (Computational Soundness). *The construction in Section 4.2 satisfies computational soundness.*

Suppose $G \notin L_{\mathsf{HAM}}$ and there exists a cheating prover $\mathcal{P}^*$ such that $\Pr[\mathcal{P}^* \text{ succeeds}] \geq 1/\lambda^c$ for infinite many λ. Then for each such λ, there must exist a $\mathbf{b}'_0$ such that $\Pr[\mathcal{P}^* \text{ succeeds} \wedge \mathbf{b}' = \mathbf{b}'_0] \geq \lambda^{-c}2^{-\mu}$, where $\mathbf{b}'$ is outputted by the cheating prover $\mathcal{P}^*$ in the second round.

$\mathbf{b}'_0$-Extractor Ext. We first describe an algorithm Ext that extracts a $\mathbf{b}'_0$ from any cheating prover $\mathcal{P}^*$, such that $\Pr[\mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}'_0] \geq \lambda^{-c}2^{-\mu-1}$. Ext receives oracle access to $\mathcal{P}^*$.

1. Initialize an empty multiset $S = \{\}$.
2. For $j \in [2^{1.5\mu}]$, set fresh random tape for $\mathcal{P}^*$. Compute and send uniformly random first round message $(\mathsf{Com}_1(1^\lambda, 1^\mu), k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}))$ to $\mathcal{P}^*$. Let $(\mathbf{b}'^{(j)}, (\mathsf{c}_i^{(j)})_{i\in[\ell]}, (z_i^{(j)})_{i\in[\ell]})$ be the response of $\mathcal{P}^*$. Execute the verifier algorithm; if verification suceeds, then append multiset $S = S \cup \{\mathbf{b}'^{(j)}\}$.
3. Output $\mathbf{b}'_0$ that appears for the maximum number of times in the multiset S.

In the sequel, we denote $p_\lambda = \Pr[\mathcal{P}^* \text{ succeeds}]$.

Lemma 5. *The algorithm* Ext *runs in time* $O(2^{1.5\mu}) = 2^{O(\log^2 \lambda)}$. *Furthermore, with probability* $1 - \exp(-\Omega(2^{0.5\mu}p_\lambda))$, *it outputs a* $\mathbf{b}'_0$ *such that* $\Pr[\mathcal{P}^* \textit{ succeeds } \wedge \mathbf{b}' = \mathbf{b}'_0] \geq p_\lambda/2^{-\mu-1}$.

We defer the proof of the Lemma 5 to the end of this proof. Now we use the extractor Ext to build the following hybrids.

Hybrid H_0 : Compute $\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*)$. Generate uniformly random string $(\mathsf{com}_1 \leftarrow \mathsf{Com}_1(1^\lambda, 1^\mu)$, $k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}))$. Send (com_1, k) to $\mathcal{P}^*$. Let $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ be the output of $\mathcal{P}^*$.

If $\mathbf{b}' = \mathbf{b}'_0$ and $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ passes the verification, then the hybrid outputs 1, otherwise outputs 0.

Hybrid H_1 : Compute $\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*)$. $\underline{\text{Generate } (\mathsf{com}_1, \mathsf{st}) \leftarrow \mathsf{FakeCom}(1^\lambda, 1^\mu, \mathbf{b}'_0)}$, $k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|})$. Send (com_1, k) to $\mathcal{P}^*$. Let $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ be the output of $\mathcal{P}^*$.

If $\mathbf{b}' = \mathbf{b}'_0$ and $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ passes the verification, then the hybrid outputs 1, otherwise output 0.

Hybrid H_2 : Compute $\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*)$. Generate $(\mathsf{com}_1, \mathsf{st}) \leftarrow \mathsf{FakeCom}(1^\lambda, 1^\mu, \mathbf{b}'_0)$, $\underline{k \leftarrow \mathsf{FakeGen}}(1^\lambda, 1^{|C_{\mathsf{st}}|}, C_{\mathsf{st}})$. Send (com_1, k) to $\mathcal{P}^*$. Let $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ be the output of $\mathcal{P}^*$.

If $\mathbf{b}' = \mathbf{b}'_0$ and $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ passes the verification, then the hybrid outputs 1, otherwise outputs 0.

This completes the description of the hybrids. We now prove Lemmas 6 and 7 to establish the indistinguishability of the hybrids.

Lemma 6. $|\Pr[\mathsf{H}_0 = 1] - \Pr[\mathsf{H}_1 = 1]| < 2^{-\Omega(\log^4 \lambda)}$.

Proof. We prove this Lemma by relying on *quasi-polynomial pseudorandom receiver's message* property of the commitment scheme (Definition 9). We build the following adversary $\mathcal{D}$ trying to distinguish the receiver's message of commitment scheme from random string.

$\mathcal{D}$ takes as input $(1^\lambda, 1^\mu, \mathsf{com}_1)$. Firstly, $\mathcal{D}$ computes $\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*)$. Then, it generates $k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|})$ and sends (com_1, k) to $\mathcal{P}^*$. Let $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ be the response of $\mathcal{P}^*$. If $\mathbf{b}' = \mathbf{b}'_0$ and $(\mathbf{b}, (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ passes the verification, then output 1. Otherwise output 0.

Now $\mathcal{D}(1^\lambda, 1^\mu, \mathsf{Com}_1(1^\lambda, 1^\mu))$ simulates the environment of H_0 for $\mathcal{P}^*$. Hence,

$$\Pr\left[\mathcal{D}(1^\lambda, 1^\mu, \mathsf{Com}_1(1^\lambda, 1^\mu)) = 1\right] = \Pr[\mathsf{H}_0 = 1]$$

Also, $\mathcal{D}(1^\lambda, 1^\mu, \mathsf{FakeCom}(1^\lambda, 1^\mu, \mathbf{b}'_0))$ simulates the environment of H_1. Hence,

$$\Pr\left[\mathcal{D}(1^\lambda, 1^\mu, \mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b}'_0)) = 1\right] = \Pr[\mathsf{H}_1 = 1]$$

From Lemma 5, $\mathcal{D}$ runs in time $2^{O(\log^2 \lambda)}$. Since the distributions $\mathsf{Com}(1^\lambda, 1^\mu)$ and $\mathsf{FakeCom}(1^\lambda, 1^\mu, \mathbf{b}'_0)$ are quasi-polynomially indistinguishable,

$$\left|\Pr\left[\mathcal{D}(1^\lambda, 1^\mu, \mathsf{Com}_1(1^\lambda, 1^\mu)) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, 1^\mu, \mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b}'_0) = 1\right]\right| < 2^{-\Omega(\log^4 \lambda)}$$

Thus, we derive that $|\Pr[\mathsf{H}_0 = 1] - \Pr[\mathsf{H}_1 = 1]| \leq 2^{-\Omega(\log^4 \lambda)}$. □

Lemma 7. $|\Pr[\mathsf{H}_1 = 1] - \Pr[\mathsf{H}_2 = 1]| < 2^{-\Omega(\log^4 \lambda)}$.

Proof. We prove this lemma by relying on *quasi-polynomial pseduorandom fake key* property of CIH. We build adversary $\mathcal{D}$ trying to distinguish the fake CIH key from uniform random string.

$\mathcal{D}$ takes as input $(1^\lambda, 1^\mu, k)$. It first computes $\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*)$. Next, it generates $\mathsf{com}_1 \leftarrow \mathsf{FakeCom}_1(1^\lambda, 1^\mu, \mathbf{b}'_0)$ and sends (com_1, k) to $\mathcal{P}^*$. Let $(\mathbf{b}', (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ be the response of $\mathcal{P}^*$. If $\mathbf{b}' = \mathbf{b}'_0$ and $(\mathbf{b}, (\mathsf{c}_i)_{i\in[\ell]}, (z_i)_{i\in[\ell]})$ passes the verification, then output 1. Otherwise output 0.

Now $\mathcal{D}(1^\lambda, 1^{|C_{\mathsf{st}}|}, k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}))$ simulates the environment of H_1 for $\mathcal{P}^*$. Hence,

$$\Pr\left[\mathcal{D}(1^\lambda, 1^{|C_{\mathsf{st}}|}, k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|})) = 1\right] = \Pr[\mathsf{H}_1 = 1]$$

Also, $\mathcal{D}(1^\lambda, 1^{|C_{\mathsf{st}}|}, k \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}, C_{\mathsf{st}}))$ simulates the environment of H_2. Hence,

$$\Pr\left[\mathcal{D}(1^\lambda, 1^{|C_{\mathsf{st}}|}, k \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}, C_{\mathsf{st}})) = 1\right] = \Pr[\mathsf{H}_2 = 1]$$

From Lemma 5, $\mathcal{D}$ runs in time $2^{O(\log^2 \lambda)}$. Since the distributions $\mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|})$ and $\mathsf{FakeGen}(1^\lambda,$
$1^{|C_{\mathsf{st}}|}, C_{\mathsf{st}})$ are quasi-polynomially indistinguishable, we have

$$\left|\Pr\left[\mathcal{D}(1^\lambda, 1^{|C_{\mathsf{st}}|}, k \leftarrow \mathsf{KGen}(1^\lambda, 1^{|C_{\mathsf{st}}|})) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, 1^{|C_{\mathsf{st}}|}, k \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}, C_{\mathsf{st}})) = 1\right]\right| < 2^{-\Omega(\log^4 \lambda)}$$

Thus, we derive $|\Pr[\mathsf{H}_1 = 1] - \Pr[\mathsf{H}_2 = 1]| \leq 2^{-\Omega(\log^4 \lambda)}$. □

We now prove the following lemma to lower bound the probability that the output of H_2 is 1.

Lemma 8. $\Pr[\mathsf{H}_2 = 1] \geq \lambda^{-c} 2^{-\mu-2} - 2 \cdot 2^{-\Omega(\log^4 \lambda)}$

Proof. From Lemma 5, we have

$$\begin{aligned}\Pr[\mathsf{H}_0 = 1] &= \Pr[\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}'_0] \\ &\geq \Pr\left[\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}'_0 \wedge \Pr[\mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}'_0] > p_\lambda 2^{-\mu-1}\right]\end{aligned}$$

$$
\begin{aligned}
&= \Pr[\mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}_0' | \Pr[\mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}_0'] > p_\lambda 2^{-\mu-1}] \\
&\cdot \Pr[\mathbf{b}_0' \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \Pr[\mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}_0'] > p_\lambda 2^{-\mu-1}] \\
&> \lambda^{-c} 2^{-\mu-1} \cdot \left(1 - \exp\left(-\Omega(2^{0.5\mu} p_\lambda)\right)\right) \geq \lambda^{-c} 2^{-\mu-2}
\end{aligned}
$$

Combining the above with the Lemma 6 and Lemma 7, we have $\Pr[\mathsf{H}_2 = 1] \geq \lambda^{-c} 2^{-\mu-2} - 2 \cdot 2^{-\Omega(\log^4 \lambda)}$. □

In the remainder of the proof, we use the $2^{-\lambda}$-correlation intractability property of the CIH to reach a contradiction. Towards this, we first show in the following lemma that $\mathsf{H}_2 = 1$ implies that there exists a 'collision' for CIH and C_{st}. Specifically, we show that any accepting proof in hybrid H_2 such that $\mathbf{b}' = \mathbf{b}_0'$, we can find a 'collision' for CIH and C_{st}.

Lemma 9. *If hybrid* H_2 *outputs* 1*, denote* $\mathsf{COM} = (\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell)$ *in the accepting proof. Then* $\mathsf{H}_k(\mathsf{COM}) = C_{\mathsf{st}}(\mathsf{COM})$*.*

Proof. We will prove by contradiction. Denote $(b_1, b_2, \ldots, b_\ell) = \mathsf{H}_k(\mathsf{COM})$. Suppose there is an $i \in [\ell]$ such that $b_i \neq C_{\mathsf{st}}(\mathsf{c}_i)$. Now we consider two cases: (1). $b_i = 0, C_{\mathsf{st}}(\mathsf{c}_i) = 1$, (2). $b_i = 1, C_{\mathsf{st}}(\mathsf{c}_i) = 0$.

For case (1), since $b_i = 0$, z_i must be of the form $(G_i, (r_i^{(s,t)})_{s,t\in[n]})$, where G_i is a cycle graph, and $\mathsf{c}_i[s,t] = \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', G_i[s,t]; r_i^{(s,t)})$ for each $s, t \in [n]$. From the extractability property of the commitment scheme and $\mathbf{b}' = \mathbf{b}_0'$, we have $\mathsf{Dec}(1^\lambda, 1^\mu, \mathsf{st}, \mathsf{c}_i[s,t]) = G_i[s,t]$. Since G_i is a cycle graph, $C_{\mathsf{st}}(\mathsf{c}_i) = 0$. Therefore, we reach a contradiction.

For case (2), since $b_i = 1$, z_i must be the form $(\phi, (r_i^{(s,t)})_{e\notin E,(s,t)=\phi(e)})$, where ϕ is a one-to-one map, and $\mathsf{c}_i[s,t] = \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', 0; r_i^{(s,t)})$ for each $e \notin E, (s,t) = \phi(e)$. Let $G_i[s,t] = \mathsf{Dec}(1^\lambda, 1^\mu, \mathsf{st}, \mathsf{c}_i[s,t])$ for each $s, t \in [n]$. Since $C_{\mathsf{st}}(\mathsf{c}_i) = 0$, G_i is a cycle graph. For each edge $e' = (s', t')$ of the cycle graph, $G_i[s', t'] = 1$. Now we will show that $(\phi^{-1}(s'), \phi^{-1}(t')) \in E$. We show this by contradiction. Suppose $(\phi^{-1}(s'), \phi^{-1}(t')) \notin E$, then $\mathsf{c}_i[s', t'] = \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', 0; r_i^{(s',t')})$. From extractable property of commitment scheme, $\mathsf{Dec}(1^\lambda, 1^\mu, \mathsf{st}, \mathsf{c}_i[s', t']) = 0$, which implies $G_i[s', t'] = 0$. Thus, we find a contradiction. Hence, for each edge e in cycle graph G_i, $\phi^{-1}(e)$ is an edge in G. Now we have found a Hamiltonian cycle $\phi^{-1}(G_i) \subseteq G$, which is a contradiction to $G \notin L_{\mathsf{HAM}}$. □

Combining Lemmas 8 and Lemma 9, we derive that

$$
\Pr\left[k \leftarrow \mathsf{FakeGen}(1^\lambda, 1^{|C_{\mathsf{st}}|}, C_{\mathsf{st}}) : \exists\ \mathsf{COM}, \mathsf{H}_k(\mathsf{COM}) = C_{\mathsf{st}}(\mathsf{COM})\right] \geq \lambda^{-c} 2^{-\mu-2} - 2 \cdot 2^{-\Omega(\log^4 \lambda)}
$$

However, the above contradicts the $2^{-\lambda}$-statistical correlation intractability of CIH.

We now finish the proof by proving Lemma 5.

Proof of Lemma 5. Extractor Ext clearly runs in time $O(2^{1.5\mu})$. To lower bound the probability $\Pr[\mathcal{P}^* \text{ succeeds } \wedge \mathbf{b}' = \mathbf{b}_0']$, we first give a lower bound on the size of multiset S. Note that in Step 2 of description of Ext, a new element is added to S with probability p_λ. From Chernoff bound,

$$
\Pr\left[|S| > 2^{1.5\mu} p_\lambda / 2\right] = 1 - \Pr\left[|S| \leq 2^{1.5\mu} p_\lambda / 2\right] \geq 1 - \exp(-2^{1.5\mu} p_\lambda / 8)
$$

From pigeonhole principle, $\mathbf{b}_0'$ outputted by Ext must appear at least $|S|/2^\mu$ times in S. Now we have

$$
\Pr\left[\mathbf{b}_0' \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \Pr[\mathcal{P}^* \text{ succeeds} \wedge \mathbf{b}' = \mathbf{b}_0'] < \frac{2^{-\mu} p_\lambda}{2}\right]
$$

$$
\begin{aligned}
&= \Pr\left[\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \Pr[\mathbf{b}' = \mathbf{b}'_0 | \mathcal{P}^* \text{ succeeds}] p_\lambda < \frac{2^{-\mu} p_\lambda}{2}\right] \\
&= \Pr\left[\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \Pr[\mathbf{b}' = \mathbf{b}'_0 | \mathcal{P}^* \text{ succeeds}] < 2^{-\mu}/2\right] \\
&\le \Pr\left[\mathbf{b}'_0 \text{ appears at least } |S|/2^\mu \text{ times in } S \wedge \Pr[\mathbf{b}' = \mathbf{b}'_0 | \mathcal{P}^* \text{ succeeds}] < 2^{-\mu}/2\right] \\
&\le \exp\left(-\frac{1}{6}|S| 2^{-\mu}\right)
\end{aligned}
$$

The last inequality follows from Chernoff bound. When $|S| \geq 2^{1.5\mu} p_\lambda / 2$, this probability is upper bounded by $\exp\left(-\frac{1}{12} 2^{0.5\mu} p_\lambda\right)$. By the union bound, we have

$$
\Pr\left[\mathbf{b}'_0 \leftarrow \mathsf{Ext}(\mathcal{P}^*) : \Pr[\mathcal{P}^* \text{ succeeds} \wedge \mathbf{b}' = \mathbf{b}'_0] < \frac{2^{-\mu} p_\lambda}{2}\right] \le \exp\left(-\frac{1}{8} 2^{1.5\mu} p_\lambda\right) + \exp\left(-\frac{1}{12} 2^{0.5\mu} p_\lambda\right)
$$

□

Theorem 8 (Statistical Witness Indistinguishability)**.** *The construction in Section 4.2 satisfies statistical witness indistinguishability. Specifically, there exists a negligible function $\nu(\lambda)$ such that for every $G \in L_{\mathsf{HAM}}$ every two witness ω_1 and ω_2 for G, every (potentially maliciously computed) fixed first round message (com_1, k), the second round prover messages Π_1 Π_2 computed using ω_1 and ω_2 respectively, satisfy*

$$
\mathsf{SD}(\Pi_1, \Pi_2) < 2^{-\Omega(\mu)} + 2n^2(\ell + 1) \cdot \nu(\lambda)
$$

We prove the theorem via a hybrid argument. For any fixed (com_1, k), let $\mathbf{b} = \mathsf{ComExt}(1^\lambda, 1^\mu, \mathsf{com}_1)$. Since $\mathbf{b}'$ is sampled uniformly at random by the prover, $\Pr[\mathbf{b} = \mathbf{b}'] = 2^{-\mu}$. Hence, with probability $1 - 2^{-\mu}$, $\mathbf{b} \neq \mathbf{b}'$. We now build a series of hybrids.

Hybrid H_0 : $(\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell) = \Pi_1$, where each z_j is computed honestly using ω_1.

Hybrid H_j : Same as above except that $z_1, z_2, \ldots, z_{j-1}$ are computed using witness ω_2, and $z_j, z_{j+1}, \ldots, z_\ell$ are computed using ω_1.

Hybrid $\mathsf{H}^1_j, (j = 0, 1, \ldots, \ell)$:

1. Sample $\mathbf{b}' \leftarrow \{0,1\}^\mu$. Generate $(\mathsf{c}_i)_{i \in [\ell] \setminus \{j\}}$ honestly in the same way as in the construction.
2. Compute $b'_j \leftarrow \{0,1\}$. Compute c_j honestly in the same way as in the construction.
3. Let $(b_1, b_2, \ldots, b_j, \ldots, b_\ell) \leftarrow \mathsf{H}_k(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_j, \ldots, \mathsf{c}_\ell)$. If $b'_j \neq b_j$, then goto 1, otherwise goto 4.
4. For $i \in [1, j-1]$, compute z_i honestly for challenge b_i using ω_2. For $i \in [j, \ell]$, compute z_i honestly for challenge b_i using ω_1. Output $(\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell)$.

Hybrid $\mathsf{H}^2_j, (j = 0, 1, \ldots, \ell)$:

1. Sample $\mathbf{b}' \leftarrow \{0,1\}^\mu$. Generate $(\mathsf{c}_i)_{i \in [\ell] \setminus \{j\}}$ honestly in the same way as in the construction.
2. Sample $b'_j \leftarrow \{0,1\}$. If $b'_j = 0$, then compute c_j honestly in the same way as in the construction, and generate z_j honestly. If $b'_j = 1$, then sample a uniformly random one-to-one map ϕ. For each $e \in \omega, (s,t) = \phi(e)$, set $G_j[s,t] = 1$. For other edges, set $G_j[s,t] = 0$. For each $s, t \in [n]$, sample uniformly random $r_j^{(s,t)}$, and compute $\mathsf{c}_j[s,t] := \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', G_j[s,t]; r_j^{(s,t)})$. Set $z_j = (\phi, (r_j^{(s,t)})_{e \notin G, (s,t) = \phi(e)})$.

3. Let $(b_1, b_2, \ldots, b_j, \ldots, b_\ell) \leftarrow \mathsf{H}_k(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_j, \ldots, \mathsf{c}_\ell)$. If $b'_j \neq b_j$, then goto 1, otherwise goto 4.
4. For $i \in [1, j-1]$, generate z_i according to the challenge b_i honestly using ω_2. For $i \in [j+1, \ell]$, generate z_i according to the challenge b_i honestly using ω_1. Output $(\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell)$.

Hybrid $\mathsf{H}_j^3, (j = 0, 1, \ldots, \ell)$:

1. Sample $\mathbf{b}' \leftarrow \{0,1\}^\mu$. Generate $(\mathsf{c}_i)_{i \in [\ell] \setminus \{j\}}$ honestly in the same way as in the construction in Section 4.2.
2. Sample $b'_j \leftarrow \{0,1\}$. If $b'_j = 0$, then compute c_j honestly in the same way as in the construction, and generate z_j honestly. If $b'_j = 1$, then sample a uniformly random one-to-one map ϕ. For each $e \notin E$, $(s,t) = \phi(e)$, sample a uniformly random $r_j^{(s,t)}$, and compute $\mathsf{c}_j[s,t] := \mathsf{Com}_2(1^\lambda, 1^\mu, \mathsf{com}_1, \mathbf{b}', G_j[s,t]; r_j^{(s,t)})$. Further, for each $e \in E$, $(s,t) = \phi(e)$, compute $\mathsf{c}_j[s,t] \leftarrow \mathsf{Sim}(1^\lambda, 1^\mu, \mathsf{com}_1)$, where Sim is the simulator for the public-coin statistical-hiding extractable commitment scheme.
3. Let $(b_1, b_2, \ldots, b_j, \ldots, b_\ell) \leftarrow \mathsf{H}_k(\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_j, \ldots, \mathsf{c}_\ell)$. If $b'_j \neq b_j$, then goto 1, otherwise goto 4.
4. For $i \in [1, j-1]$, generate z_i according to the challenge b_i honestly using ω_2. For $i \in [j+1, \ell]$, generate z_i according to the challenge b_i honestly using ω_1. Output $(\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell)$.

Hybrid $\mathsf{H}_{\ell+1}$: $(\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell) = \Pi_2$, where each z_j are generated using ω_2.

This completes the description of the hybrids. We now prove a series of lemmas to bound the statistical distance between different adjacent hybrids. The proof then follows by combining the claims of the lemmas.

Lemma 10. $\mathsf{SD}(\mathsf{H}_j, \mathsf{H}_j^1) = 0$

Proof. The difference between H_j and H_j^1 is that H_j^1 has a rejection sampling process on b_j. Hence, we have

$$\begin{aligned} \Pr\left[\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell \mid \mathsf{H}_j^1\right] &= \Pr\left[\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell \mid \mathsf{H}_j, b_j = b'_j\right] \\ &= \Pr\left[\mathbf{b}', \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_\ell, z_1, z_2, \ldots, z_\ell \mid \mathsf{H}_j\right] \end{aligned}$$

The second equality comes from the fact that b'_j is chosen uniformly at random. □

Lemma 11. $\mathsf{SD}\left(\mathsf{H}_j^1, \mathsf{H}_j^2\right) = 0$

Proof. The only difference between H_j^1 and H_j^2 is that in H_j^1, we sample a cycle graph G_j uniformly at random whereas in H_j^2, we first sample the one-to-one map ϕ uniformly at random and then generate $G_i = \phi(w)$. Hence, the distributions over (ϕ, G_i) in H_j^1 and H_j^2 are identical. □

Lemma 12. $\mathsf{SD}\left(\mathsf{H}_j^2, \mathsf{H}_j^3\right) < n^2 \cdot \nu(\lambda) + 2^{-\Omega(\mu)}$

Proof. The difference between H_j^2 and H_j^3 is that in H_j^3, we use the simulator of the public-coin statistical-hiding commitment scheme for computing $\mathsf{c}_i[s,t]$ for each $e \in E$, $(s,t) = \phi(e)$. However, since the randomness $r_j^{(s,t)}$ for each such $\mathsf{c}_i[s,t]$ is never opened, the lemma follows from the statistical hiding property of the commitment scheme. □

Lemma 13. $\mathsf{SD}\left(\mathsf{H}_j^3, \mathsf{H}_{j+1}\right) < n^2 \cdot \nu(\lambda) + 2^{-\Omega(\mu)}$

Proof. Note that proving $\mathsf{SD}(\mathsf{H}_j^3, \mathsf{H}_{j+1}) < n^2 \cdot \nu(\lambda) + 2^{-\Omega(\mu)}$ is symmetric to proving that $\mathsf{SD}(\mathsf{H}_j, \mathsf{H}_j^3) < n^2 \cdot \nu(\lambda) + 2^{-\Omega(\mu)}$. The latter follows by combining Lemmas 10, 11, 12. The proof follows the same strategy as previous lemmas. □

5 Statistical Hash Commitments

Intuitively speaking, a statistical hash commitment (SHC) scheme is a two-round statistically hiding commitment scheme, where the verification of the decommitment is a simple equality check with a hash output (computed w.r.t. a hashing algorithm associated with the scheme).

Definition 10. *A statistical hash commitment scheme is a tuple of algorithms* $(\mathsf{KGen}, \mathsf{Com}, \mathsf{H}, \mathcal{C}, \mathcal{R})$*. It proceeds as follows.*

Round 1 $\mathcal{R}$ *executes* $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda)$*, and sends* pk *to* $\mathcal{C}$*.*

Round 2 $\mathcal{C}$*'s input is a bit* $b \in \{0,1\}$*. Compute* $(c, \rho) \leftarrow \mathsf{Com}(\mathsf{pk}, b)$ *and send* c *to* $\mathcal{R}$*.*

Opening $\mathcal{C}$ *sends* (b, ρ) *to the* $\mathcal{R}$*.*

Verification $\mathcal{R}$ *accepts iff* ρ *is equal to* $\mathsf{H}(\mathsf{k}, c, b)$*.*

We require the scheme to satisfy the following properties.

Completeness For any $b \in \{0,1\}$, we have

$$\Pr\left[(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda), (c, \rho) \leftarrow \mathsf{Com}(\mathsf{pk}, b) : \rho = \mathsf{H}(\mathsf{k}, c, b)\right] = 1$$

Computational Binding We say that the commitment scheme is computational binding, if for any non-uniform probabilistic polynomial time adversary $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that

$$\mathsf{Adv}(\mathcal{A}) \triangleq \Pr\left[(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda), (c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk}) : {\scriptstyle \rho_0 = \mathsf{H}(\mathsf{k}, c, 0) \wedge \atop \rho_1 = \mathsf{H}(\mathsf{k}, c, 1)}\right] < \nu(\lambda)$$

Statistical Hiding For any (maliciously generated) pk, there exists a negligible function $\nu(\lambda)$ such that $\mathsf{SD}(c_0, c_1) \le \nu(\lambda)$, where $(c_b, \rho_b) \leftarrow \mathsf{Com}(\mathsf{pk}, b)$ for every $b \in \{0,1\}$. If $\nu(\lambda) = 0$, then we say that the scheme is perfectly hiding.

5.1 Construction from CDH

Let q be an integer, and $G = \langle g \rangle$ be a cyclic group generated by g of order q.

Construction. We describe our construction of the SHC scheme.

$\mathsf{KGen}(1^\lambda)$ Randomly sample $s, t \leftarrow \mathbb{Z}_q$, and $x \leftarrow G$. Output $\left(\mathsf{pk} = (x, g^s, x^s \cdot g^t), \mathsf{k} = (s, t)\right)$.

$\mathsf{Com}(\mathsf{pk}, b)$ Parse pk as $(x, a_1, a_2) \in G \times G$. Randomly sample $u, v \leftarrow \mathbb{Z}_q$.
Output $\left(c = (g^u \cdot x^v, g^v \cdot g^b), \rho = a_1^u \cdot a_2^v\right)$.

$\mathsf{H}(\mathsf{k}, c, b)$ Parse c as $(z_1, z_2) \in G \times G$, and parse k as (s, t). Output $z_1^s \cdot (z_2 \cdot g^{-b})^t$.

We now prove the properties of this construction.

Lemma 14 (Completeness)**.** *The construction above satisfies completeness.*

Proof. For any $b \in \{0,1\}$, $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda)$, where $\mathsf{pk} = (x, g^s, x^s \cdot g^t), \mathsf{k} = (s,t)$. Let $(c, \rho) \leftarrow \mathsf{Com}(\mathsf{pk}, b)$. Then $c = (g^u \cdot x^v, g^v \cdot g^b)$, and $\rho = g^{su+tv} \cdot x^{sv}$. Hence, $\mathsf{H}(\mathsf{k}, c, b) = (g^u \cdot x^v)^s \cdot (g^v)^t = \rho$. □

Lemma 15 (Computational Binding)**.** *Assuming CDH, the above construction of SHC is computational binding.*

Proof. For any n.u. probabilistic polynomial time adversary $\mathcal{A}$, we construct the following adversary $\mathcal{A}'$ for CDH problem.

Adversary $\mathcal{A}'(1^\lambda, g^s, g^y)$ Sample $u \leftarrow \mathbb{Z}_q$ uniformly at random. Set $x = g^y, \mathsf{pk} = (x, g^s, g^u)$. Execute $(c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$. Output $g^u \cdot \rho_0^{-1} \cdot \rho_1$.

We now prove that $\Pr[a \leftarrow \mathcal{A}'(1^\lambda, g^s, g^y) : a = g^{sy}] \geq \mathsf{Adv}(\mathcal{A})$. Since in our construction, $\mathsf{pk} = (x, g^s, x^s \cdot g^t)$, where t is uniformly random. The second component of pk is uniformly random over G. Hence, the distributions of pk in real execution and the adversary $\mathcal{A}'$ are identical.

Now for any $u \in \mathbb{Z}_q$, there exists an unique $t' \in \mathbb{Z}_q$ such that $x^s \cdot g^{t'} = g^u$. Then, for adversary $\mathcal{A}'$, we have

$$\begin{aligned}\Pr\left[a = g^{sy}\right] &= \Pr\left[g^u \cdot \rho_0^{-1} \cdot \rho_1 = g^{sy}\right] = \Pr\left[g^{t'} = \rho_0 \cdot \rho_1^{-1}\right] \\ &\geq \Pr\left[\rho_0 = \mathsf{H}(\mathsf{k}, c, 0) \wedge \rho_1 = \mathsf{H}(\mathsf{k}, c, 1)\right] = \mathsf{Adv}(\mathcal{A})\end{aligned}$$

where $\mathsf{k} = (s, t')$. By the hardness of CDH, we conclude that $\mathsf{Adv}(\mathcal{A})$ is negligible. □

Lemma 16 (Perfect Hiding)**.** *The Construction 5.1 is perfect hiding.*

Proof. For any fixed $\mathsf{pk} = (x, a_1, a_2)$, since v is uniformly random, $g^v \cdot g^b$ is uniformly random. Furthermore, conditioned on $g^v \cdot g^b$, since u is uniformly random, $g^u \cdot x^v$ is also uniformly random. Hence, c is uniformly random over $G \times G$. □

5.2 Construction from any 2-round Statistical Sender-Private OT

We now describe our construction of SHC from statistical sender-private OT. Let $\ell = \ell(\lambda)$ be a polynomial in λ, and let $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3)$ be any statistical sender private 2-round OT scheme.

$\mathsf{KGen}(1^\lambda)$ Randomly sample $r \leftarrow \{0,1\}^\ell$.

For $i \in [\ell]$, execute $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, r[i])$.

Output $\mathsf{pk} = ((\mathsf{ot}_{1,i})_{i \in [\ell]}, \mathsf{k} = (\mathsf{st}_i)_{i \in [\ell]})$.

$\mathsf{Com}(\mathsf{pk}, b \in \{0,1\})$ Parse pk as $(\mathsf{ot}_{1,i})_{i \in [\ell]}$. Randomly sample $r' \leftarrow \{0,1\}^\ell$.

For $i \in [\ell]$, execute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, r'[i], r'[i] \oplus b)$.

Output $(c = (\mathsf{ot}_{2,i})_{i \in [\ell]}, \rho = r')$.

$\mathsf{H}(\mathsf{k}, c, b)$ Parse $\mathsf{k} = (\mathsf{st}_i)_{i \in [\ell]}, c = (\mathsf{ot}_{2,i})_{i \in [\ell]}$.

For $i \in [\ell]$, Let $\rho_{0,i} \leftarrow \mathsf{OT}_3(\mathsf{st}_i, \mathsf{ot}_{2,i})$.

Let $\rho_b = (\rho_{0,i} \oplus (r[i] \cdot b))_{i \in [\ell]}$.

Output ρ_b.

Lemma 17 (Completeness)**.** *The above construction of SHC is complete.*

Proof. For any $b \in \{0, 1\}$, let $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda)$, $(c, \rho) \leftarrow \mathsf{Com}(\mathsf{pk}, b)$. From the construction of the commitment, we know that $\rho_{0,i} = r'[i] \oplus (r[i] \cdot b)$. From the construction of $\mathsf{H}(\mathsf{k}, c, b)$, we have $\rho_{b,i} = \rho_{0,i} \oplus (r[i] \cdot b) = (r'[i] \oplus (r[i] \cdot b)) \oplus (r[i] \cdot b) = r'[i] = \rho$. Hence, the opening (b, ρ) is accepted. □

Lemma 18 (Computational Binding)**.** *Assuming computational indistinguishability of* OT_1*, the above construction of SHC is computational binding.*

Proof. For any PPT adversary $\mathcal{A}$ trying to break the computational binding property, we construct the following hybrids.

Hybrid H_0 Randomly sample $r \leftarrow \{0, 1\}^\ell$. For $i \in [\ell]$, execute $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, r[i])$. Let $\mathsf{pk} = (\mathsf{ot}_{1,i})_{i \in [\ell]}$. Execute $(c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$. If $\rho_0 \oplus \rho_1 = r$, then output 1, otherwise output 0.

Hybrid $\mathsf{H}_{0.5}^{i^*}$ Randomly sample $r \leftarrow \{0, 1\}^\ell$. For $1 \le i \le i^*$, execute $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, 0)$. For $i^* < i \le \ell$, execute $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, r[i])$. Let $\mathsf{pk} = (\mathsf{ot}_{1,i})_{i \in [\ell]}$. Execute $(c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$. If $\rho_0 \oplus \rho_1 = r$, then output 1, otherwise output 0.

Hybrid H_1 Randomly sample $r \leftarrow \{0, 1\}^\ell$. For $i \in [\ell]$, execute $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, 0)$. Let $\mathsf{pk} = (\mathsf{ot}_{1,i})_{i \in [\ell]}$. Execute $(c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$. If $\rho_0 \oplus \rho_1 = r$, then output 1, otherwise output 0.

Lemma 19. $\Pr[\mathsf{H}_0 = 1] \ge \mathsf{Adv}(\mathcal{A})$.

Proof. From the construction of H, we now that $\mathsf{H}(\mathsf{k}, c, 0) \oplus \mathsf{H}(\mathsf{k}, c, 1) = r$. Hence, when $\mathcal{A}$ wins the security game, $(c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$ with $\rho_0 = \mathsf{H}(\mathsf{k}, x, 0) \wedge \rho_1 = \mathsf{H}(\mathsf{k}, x, 1)$ implies $\rho_0 \oplus \rho_1 = \mathsf{H}(\mathsf{k}, x, 0) \oplus \mathsf{H}(\mathsf{k}, x, 1) = r$. □

Lemma 20. *Hybrid* H_0 *and Hybrid* $\mathsf{H}_{0.5}^0$ *are identical. Furthermore, there exists a negligible function* $\nu(\lambda)$ *such that for each* $i = 0, \ldots, \ell - 1$, $|\Pr[\mathsf{H}_{0.5}^{i^*} = 1] - \Pr[\mathsf{H}_{0.5}^{i^*+1} = 1]| < \nu(\lambda)$.

Proof. When $i^* = 0$, all $\mathsf{ot}_{1,i}$ are generated in the same way as in Hybrid H_0, for all $i \in [\ell]$. Hence, Hybrid H_0 and Hybrid $\mathsf{H}_{0.5}^0$ are identical.

To show $\mathsf{H}_{0.5}^{i^*} \approx \mathsf{H}_{0.5}^{i^*+1}$, we consider the following adversary $\mathcal{D}$ for receiver's computational privacy.

$\mathcal{D}(1^\lambda, \mathsf{ot}_1)$ Randomly sample $r \leftarrow \{0, 1\}^\ell$. For $i \in [\ell] \setminus \{i^* + 1\}$, let $(\mathsf{ot}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{OT}_1(1^\lambda, r[i])$. If $r[i^* + 1] = 0$, then let $(\mathsf{ot}_{1,i^*+1}, \mathsf{st}_{i^*+1}) \leftarrow \mathsf{OT}_1(1^\lambda, 0)$, otherwise let $\mathsf{ot}_{1,i^*+1} = \mathsf{ot}_1$. Let $\mathsf{pk} = (\mathsf{ot}_{1,i})_{i \in [\ell]}$. Execute $(c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$. If $\rho_0 \oplus \rho_1 = r$, then output 1, otherwise output 0.

If ot_1 is generated from $\mathsf{OT}_1(1^\lambda, 0)$, then $\mathcal{D}$ simulates the environment of $\mathsf{H}_{0.5}^{i^*+1}$ for $\mathcal{A}$. Hence,

$$\Pr\left[\mathsf{H}_{0.5}^{i^*+1} = 1\right] = \Pr\left[(\mathsf{ot}_1, \mathsf{st}) \leftarrow \mathsf{OT}_1(1^\lambda, 0) : \mathcal{D}(1^\lambda, \mathsf{ot}_1) = 1\right]$$

If ot_1 is generated from $\mathsf{OT}_1(1^\lambda, 1)$, then $\mathcal{D}$ simulates the environment of $\mathsf{H}_{0.5}^{i^*}$ for $\mathcal{A}$. Hence,

$$\Pr\left[\mathsf{H}_{0.5}^{i^*} = 1\right] = \Pr\left[(\mathsf{ot}_1, \mathsf{st}) \leftarrow \mathsf{OT}_1(1^\lambda, 1) : \mathcal{D}(1^\lambda, \mathsf{ot}_1) = 1\right]$$

From the indistinguishability of ot_1, we know that the right hand ot_1^0 generated by $\mathsf{OT}_1(1^\lambda, 0)$ and ot_1^1 generated by $\mathsf{OT}_1(1^\lambda, 1)$ are indistinguishable. Hence, there exists a negligible function $\nu(\lambda)$ such that $|\Pr[\mathsf{H}_{0.5}^{i^*} = 1] - \Pr[\mathsf{H}_{0.5}^{i^*+1} = 1]| < \nu(\lambda)$. □

Lemma 21. *Hybrid* $\mathsf{H}_{0.5}^{\ell}$ *is identical to* H_1*. Furthermore,* $\Pr[\mathsf{H}_1 = 1] = 1/2^{\ell}$*.*

Proof. When $i^* = \ell$, we know that all $\mathsf{ot}_{1,i}$ are generated in the same way as in Hybrid H_1. Hence, $\mathsf{H}_{0.5}^{\ell}$ and H_1 are identical.

In Hybrid H_1, pk is completely independent of r. Hence, $\Pr[\mathsf{H}_1 = 1] = \Pr[\rho_0 \oplus \rho_1 = r] = 1/2^{\ell}$. □

By the hybrid argument, combining Lemma 19, Lemma 20, and Lemma 21, we have $\mathsf{Adv}(\mathcal{A}) < \mathsf{neg}(\lambda)$. □

Lemma 22 (Statistical Hiding)**.** *Assuming the underlying OT scheme is statistical sender private, the above construction of SHC is statistical hiding.*

Proof. For any (maliciously generated) pk, and any bit $b \in \{0,1\}$, we build the following hybrids.

Hybrid H_0 Randomly sample $r' \leftarrow \{0,1\}^{\ell}$.

For $i \in [\ell]$, execute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, r'[i], r'[i] \oplus b)$.

Output $c = (\mathsf{ot}_{2,i})_{i \in [\ell]}$.

Hybrid $\mathsf{H}_{0.5}^{i^*}$ Randomly sample $r' \leftarrow \{0,1\}^{\ell}$.

For $1 \le i \le i^*$, execute $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, r'[i], r'[i])$.

For $i^* < i \le \ell$, randomly sample $r''_i \leftarrow \{0,1\}$, let $\mathsf{ot}_{2,i} \leftarrow \mathsf{OT}_2(\mathsf{ot}_{1,i}, r''_i, r''_i)$.

Output $c = (\mathsf{ot}_{2,i})_{i \in [\ell]}$.

Lemma 23. *Hybrid* H_0 *is identical to* $\mathsf{H}_{0.5}^{0}$*. Furthermore, there exists a negligible function* $\nu(\lambda)$ *such that, for any* $i^* = 0, \ldots, \ell - 1$*, we have* $\mathsf{SD}(\mathsf{H}_{0.5}^{i^*}, \mathsf{H}_{0.5}^{i^*+1}) < \nu(\lambda)$*.*

Proof. For each $i^* = 0, \ldots, \ell - 1$, let $b_{i^*+1} \leftarrow \mathsf{OTExt}(\mathsf{ot}_{1,i^*+1})$. Then $\mathsf{SD}(\mathsf{OT}_2(\mathsf{ot}_{1,i^*+1}, r'[i^*+1], r'[i] \oplus b), \mathsf{OT}_2(\mathsf{ot}_{1,i^*+1}, r'[i] \oplus (b \cdot b_{i^*+1}), r'[i] \oplus (b \cdot b_{i^*+1}))) < \nu(\lambda)$. Let $r''_i = r'[i] \oplus (b \cdot b_{i^*+1})$. Then r''_i is independent of b, and is uniformly random over $\{0,1\}$. Hence, Hybrid $\mathsf{H}_{0.5}^{i^*}$ is statistically close to $\mathsf{H}_{0.5}^{i^*+1}$. □

From Lemma 23, H_0 is statistically close to $\mathsf{H}_{0.5}^{\ell}$, regardless of b. Since $\mathsf{H}_{0.5}^{\ell}$ is independent of b, we have that c_0 and c_1 are statistically close, where c_b is generated by $\mathsf{Com}(\mathsf{pk}, b)$. □

6 Three Round Statistical Receiver-Private Oblivious Transfer

We start by presenting the definition for 3-round statistical receiver-private oblivious transfer. We capture statistical receiver privacy via a game-based definition. We consider two definitions to capture computational sender privacy: a game-based definition that intuitively requires that any malicious receiver who interacts with an honest sender can only learn one of its two inputs, and a distinguisher-dependent simulation based definition. We prove that the game-based security for senders implies distinguisher-dependent simulation security.

Definition 11 (3-round Statistical Receiver-Private Oblivious Transfer)**.** *A 3-round oblivious transfer is a tuple of algorithms* $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$*, which specify the following protocol.*

Round 1 *The sender* $\mathcal{S}$ *computes* $(\mathsf{ot}_1, \mathsf{st}_S) \leftarrow \mathsf{OT}_1(1^{\lambda})$ *and sends* ot_1 *to the receiver* $\mathcal{R}$*.*

Round 2 *The receiver* $\mathcal{R}$ *with input* $\beta \in \{0,1\}$*, computes* $(\mathsf{ot}_2, \mathsf{st}_R) \leftarrow \mathsf{OT}_2(1^{\lambda}, \mathsf{ot}_1, \beta)$*. Send* ot_2 *to* $\mathcal{S}$*.*

Round 3 *$\mathcal{S}$ with input $(m_0, m_1) \in \{0,1\}^2$ computes $\mathsf{ot}_3 \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st}_S, m_0, m_1)$. Send ot_3 to $\mathcal{R}$.*

Message Decryption *The receiver computes $m' \leftarrow \mathsf{OT}_4(1^\lambda, \mathsf{ot}_1, \mathsf{ot}_3, \mathsf{st}_R)$.*

We require the protocol to satisfy the following properties.

Correctness [3] *For any $\beta \in \{0,1\}, (m_0, m_1) \in \{0,1\}^2$, we have*

$$\Pr\left[\begin{smallmatrix} (\mathsf{ot}_1, \mathsf{st}_S) \leftarrow \mathsf{OT}_1(1^\lambda) \\ (\mathsf{ot}_2, \mathsf{st}_R) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, \beta) \\ \mathsf{ot}_3 \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st}_S, m_0, m_1) \\ m' \leftarrow \mathsf{OT}_4(1^\lambda, \mathsf{ot}_1, \mathsf{ot}_3, \mathsf{st}_R) \end{smallmatrix} : m' = m_\beta\right] = 1$$

Game-Based Statistical Receiver-Privacy *For any (potentially maliciously generated) ot_1^*, denote $(\mathsf{ot}_2^{(0)}, \mathsf{st}_R^{(0)}) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1^*, 0)$, and $(\mathsf{ot}_2^{(1)}, \mathsf{st}_R^{(1)}) \leftarrow \mathsf{OT}_2(1^\lambda, \mathsf{ot}_1^*, 1)$. Then we have $\mathsf{SD}(\mathsf{ot}_2^{(0)}, \mathsf{ot}_2^{(1)}) < \nu(\lambda)$, where $\nu(\cdot)$ is a negligible function.*

Game-Based Computational Sender-Privacy *For any probabilistic polynomial time distinguisher $\mathcal{A}_0, \mathcal{A}_1$, and any probabilistic polynomial time malicious receiver $\mathcal{R}^*$, we define the following games.*

Interact with $\mathcal{R}^*$ *The challenger plays the role of an honest sender for the first round and the second round with the malicious receiver $\mathcal{R}^*$. Specifically, the challenger executes $(\mathsf{ot}_1, \mathsf{st}_S) \leftarrow \mathsf{OT}_1(1^\lambda)$. Then send ot_1 to $\mathcal{R}^*$. Then the receiver $\mathcal{R}^*$ sends ot_2^* to the challenger.*

Game $\mathsf{G}_0(m_0, m_1)$ *This game interact with adversary $\mathcal{A}_0$. In the beginning, the adversary $\mathcal{A}_0$ is given input $\mathsf{View}(\mathcal{R}^*)$. Then the challenger samples $b_0 \leftarrow \{0,1\}$ at random, and send $\mathsf{ot}_3 \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_b, m_1)$ to $\mathcal{A}_0$. Finally $\mathcal{A}_0$ outputs a bit b_0'. If $b_0 = b_0'$, then we say $\mathcal{A}_0$ wins the game.*

Game $\mathsf{G}_1(m_0, m_1)$ *This game interact with adversary $\mathcal{A}_1$. In the beginning, the adversary $\mathcal{A}_1$ is given input $\mathsf{View}(\mathcal{R}^*)$. Then the challenger samples $b_1 \leftarrow \{0,1\}$ at random, and send $\mathsf{ot}_3 \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_0, m_b)$ to $\mathcal{A}_1$. Finally $\mathcal{A}_1$ outputs a bit b_1'. If $b_1 = b_1'$, then we say $\mathcal{A}_1$ wins the game.*

We define the following advantage

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{R}^*) \triangleq \mathbb{E}_{\mathsf{View}(\mathcal{R}^*)}\Bigg[\min\Bigg\{ & \max_{m_0, m_1 \in \{0,1\}} \left(\left|\Pr[\mathcal{A}_0(\mathsf{View}(\mathcal{R}^*)) \text{ wins } \mathsf{G}_0(m_0, m_1)] - \frac{1}{2}\right|\right), \\ & \max_{m_0, m_1 \in \{0,1\}} \left(\left|\Pr[\mathcal{A}_1(\mathsf{View}(\mathcal{R}^*)) \text{ wins } \mathsf{G}_1(m_0, m_1)] - \frac{1}{2}\right|\right)\Bigg\}\Bigg]\end{aligned}$$

We say the oblivious transfer scheme is game-based computational sender-secure, if for any probabilistic polynomial time distinguisher $\mathcal{A}_0, \mathcal{A}_1$, and any probabilistic polynomial time malicious receiver $\mathcal{R}^*$, there exist a negligible function $\nu(\cdot)$ such that $\mathsf{Adv}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{R}^*) < \nu(\lambda)$.

Distinguisher-Dependent Sender-Privacy *We define distinguisher-dependent simulation-based security for senders in the real-ideal paradigm. Let $\mathcal{F}_{OT}$ be the ideal functionality of the oblivious transfer. In the ideal world, a simulator OTSim submits a bit β' to $\mathcal{F}_{OT}$, the*

[3] We can relax the definition to be statistical correctness, which only requires the probability to be $1 - \mathsf{negl}(\lambda)$

sender sends its inputs (m_0, m_1) to $\mathcal{F}_{OT}$. Then $\mathcal{F}_{OT}$ sends $m_{\beta'}$ to OTSim. Finally OTSim outputs the view of the malicious receiver $\mathcal{R}^*$ that is computationally indistinguishable with view of the receiver in the real execution.

We say that an oblivious transfer scheme is distinguisher-dependent sender-private, if for any malicious receiver $\mathcal{R}^*$, any probabilistic polynomial time distinguisher $\mathcal{D}$, and any function $\epsilon = 1/\mathsf{poly}(\lambda)$, there exists a simulator $\mathsf{OTSim}_{\mathcal{R}^*,\mathcal{D}}$ that runs in $\mathsf{poly}(1/\epsilon)$ time, and interact with the ideal functionality $\mathcal{F}_{OT}$ such that

$$\left|\Pr\left[\mathcal{D}(1^\lambda, \mathsf{OTSim}^{\mathcal{F}_{OT}}_{\mathcal{R}^*,\mathcal{D}}) = 1\right] - \Pr\left[\mathcal{D}(1^\lambda, \mathsf{View}(\mathcal{R}^*)) = 1\right]\right| < \epsilon + \mathsf{negl}(\lambda)$$

We prove the following lemma that establishes that game-based computational sender-privacy implies distinguisher-dependent sender-privacy.

Lemma 24. *If a 3-round oblivious transfer scheme $(\mathsf{OT}_1, \mathsf{OT}_2, \mathsf{OT}_3, \mathsf{OT}_4)$ satisfies game-based computational sender-privacy, then it satisfies distinguisher-dependent sender-privacy.*

Proof. For any malicious receiver $\mathcal{R}^*$ and distinguisher $\mathcal{D}$, we firstly construct an adversary $(\mathcal{A}_0, \mathcal{A}_1, \mathcal{R}^*_2)$ in the game-based computational sender-privacy.

Adversary $\mathcal{R}^*_2$ It firstly sets random coins for $\mathcal{R}^*$. Then it receives the first round message ot_1 from the challenger, it forwards the message ot_1 to the malicious receiver $\mathcal{R}^*$, and then $\mathcal{R}^*$ outputs ot^*_2. Finally $\mathcal{R}^*_2$ forwards the second round message ot^*_2 to the challenger.

Distinguisher $\mathcal{A}_0, \mathcal{A}_1$ Let $\mathcal{A}_0$ (resp. $\mathcal{A}_1$) be the following distinguisher. Given the input $\mathsf{View}(\mathcal{R}^*_2)$, which contains the random coins for $\mathcal{R}^*$ and the transcripts $\mathsf{ot}_1, \mathsf{ot}^*_2$. Set the random coins and execute $\mathcal{R}^*$. Feed the first round message ot_1 to $\mathcal{R}^*$, and $\mathcal{R}^*$ outputs its second round message, and waits for the third round message. Now the distinguisher $\mathcal{A}_0$ (resp. $\mathcal{A}_1$) interacts with the challenger in game G_0 (resp. G_1). The challenger sends the message ot_3. The adversary $\mathcal{A}_0$ (resp. $\mathcal{A}_1$) forwards the message ot_3 to $\mathcal{R}^*$. Finally $\mathcal{A}_0$ executes $\mathcal{D}$ with input $\mathsf{View}(\mathcal{R}^*)$, and outputs the output of $\mathcal{D}$.

Simulator $\mathsf{OTSim}_{\mathcal{R}^*,\mathcal{D}}$ The simulator executes the adversary $\mathcal{R}^*_2$ first, then runs the following estimator for any $(m_0, m_1) \in \{0,1\}^2$.

Estimator(m_0, m_1) Execute the following for $N = 1/\epsilon^3$ time. Set the random coins for $\mathcal{A}_0, \mathcal{A}_1$, and execute the game $\mathsf{G}_0(m_0, m_1), \mathsf{G}_1(m_0, m_1)$ for $\mathcal{A}_0(\mathsf{View}(\mathcal{R}^*_2))$, $\mathcal{A}_1(\mathsf{View}(\mathcal{R}^*_2))$ respectively. For any $b \in \{0,1\}$, let $N^b_{m_0,m_1}$ be the number of times that $\mathcal{A}_b$ wins. Output $(\tilde{p}^b_{m_0,m_1} = N^b_{m_0,m_1}/N)_{b\in\{0,1\}}$.

If $\max_{m_0,m_1\in\{0,1\}}(\tilde{p}^0_{m_0,m_1}) > \max_{m_0,m_1\in\{0,1\}}(\tilde{p}^1_{m_0,m_1})$, then query $\mathcal{F}_{OT}$ with $\beta' = 0$, otherwise query $\mathcal{F}_{OT}$ with $\beta' = 1$. Next, the ideal functionality $\mathcal{F}_{OT}$ replies with $m_{\beta'}$. The simulator then sends $\mathsf{ot}_3 \leftarrow \mathsf{OT}_3(1^\lambda, \mathsf{ot}^*_2, \mathsf{st}_S, m_{\beta'}, m_{\beta'})$ to $\mathcal{R}^*$. Finally the simulator outputs the view of $\mathcal{R}^*$.

We now show $|\Pr[\mathcal{D}(1^\lambda, \mathsf{OTSim}_{\mathcal{R}^*,\mathcal{D}}) = 1] - \Pr[\mathcal{D}(1^\lambda, \mathsf{View}(\mathcal{R}^*)) = 1]| < \epsilon + \mathsf{negl}(\lambda)$. Denote $\delta = \frac{1}{2}\max(\epsilon/2 - \sqrt{\nu(\lambda)}, 0)$, and $p^b_{m_0,m_1} = \Pr[\mathcal{A}_b(\mathsf{View}(\mathcal{R}^*_2)) \text{ wins } \mathsf{G}_b(m_0, m_1))]$. From Hoeffding's inequality, with probability $1 - 16\exp(-2N\delta^2)$, we have

$$|\tilde{p}^b_{m_0,m_1} - p^b_{m_0,m_1}| < \delta,\ \forall\ b, m_0, m_1 \in \{0,1\} \tag{2}$$

Since the scheme is game-based computational sender-private, by Markov inequality, we have

$$\Pr\left[\mathsf{v} \leftarrow \mathsf{View}(\mathcal{R}^*_2) : \forall b \in \{0,1\}, \max_{m_0,m_1}\left(\left|\Pr[\mathcal{A}_b(\mathsf{v}) \text{ wins } \mathsf{G}_b(m_0, m_1)] - \frac{1}{2}\right|\right) \geq \sqrt{\nu(\lambda)}\right] \leq \sqrt{\nu(\lambda)}$$

Hence, at least $1-\sqrt{\nu(\lambda)}$ fraction of $\mathsf{View}(\mathcal{R}_2^*)$ satisfies

$$\exists\, b \in \{0,1\}, \forall\, (m_0,m_1) \in \{0,1\}^2, \left|\Pr[\mathcal{A}_b(\mathsf{v}) \text{ wins } \mathsf{G}_b(m_0,m_1)] - \frac{1}{2}\right| < \sqrt{\nu(\lambda)}$$

Denote these kind of $\mathsf{View}(\mathcal{R}_2^*)$ as GOOD. For any fixed $\mathsf{v} \in \mathsf{GOOD}$, if $\beta' \neq b$, then in game G_b, the adversary $\mathcal{A}_b$ is asked to distinguish $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_0, m_1)$ from $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_{1-b}, m_{1-b})$. The later is identical to $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_{\beta'}, m_{\beta'})$, which is outputted by the simulator OTSim. Hence, we can bound the advantage of $\mathcal{D}$.

$$\left|\Pr\left[\mathcal{D}(1^\lambda, \mathsf{OTSim}_{R^*,\mathcal{D}}) = 1 \mid \mathsf{v} = \mathsf{View}(\mathcal{R}_2^*)\right] - \Pr\left[\mathcal{D}(1^\lambda, \mathsf{View}(\mathcal{R}^*)) = 1 \mid \mathsf{v} = \mathsf{View}(\mathcal{R}_2^*)\right]\right| < 2\sqrt{\nu(\lambda)}$$

If $\beta' = b$, denote $Q = \max_{m_0,m_1\in\{0,1\}} \left|\Pr[\mathcal{A}_{\beta'} \text{ wins } \mathsf{G}_{\beta'}(m_0,m_1)] - \frac{1}{2}\right|$. We will bound $\max_{m_0,m_1\in\{0,1\}}(p^{1-\beta'}_{m_0,m_1})$. From the choice of β', we have $\max_{m_0,m_1\in\{0,1\}}(\tilde{p}^{\beta'}_{m_0,m_1}) > \max_{m_0,m_1\in\{0,1\}}(\tilde{p}^{1-\beta'}_{m_0,m_1})$. Then from Equation 2, we have

$$\max_{m_0,m_1}(p^{1-\beta'}_{m_0,m_1}) < \max_{m_0,m_1}(\tilde{p}^{1-\beta'}_{m_0,m_1}) + \delta < \max_{m_0,m_1}(\tilde{p}^{\beta'}_{m_0,m_1}) + \delta < \max_{m_0,m_1}(p^{\beta'}_{m_0,m_1}) + 2\delta < \sqrt{\nu(\lambda)} + 2\delta \leq \frac{\epsilon}{2}$$

Since in the game $\mathsf{G}_{1-\beta'}$, the distinguisher is asked to distinguish $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_0, m_1)$ and $\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2^*, \mathsf{st}_S, m_{\beta'}, m_{\beta'})$. Now we can bound the advantage of $\mathcal{D}$ by ϵ.

$$\left|\Pr\left[\mathcal{D}(1^\lambda, \mathsf{OTSim}_{R^*,\mathcal{D}}) = 1 \mid \mathsf{v} = \mathsf{View}(\mathcal{R}_2^*)\right] - \Pr\left[\mathcal{D}(1^\lambda, \mathsf{View}(\mathcal{R}^*)) = 1 \mid \mathsf{v} = \mathsf{View}(\mathcal{R}_2^*)\right]\right| < \epsilon$$

Summary the two situations, the advantage of $\mathcal{D}$ is $2\sqrt{\nu(\lambda)}+\epsilon+16\exp(-2N\delta^2) < \epsilon+\mathsf{negl}(\lambda)$. □

6.1 Our Construction

We now describe a generic transformation from an SHC scheme to three-round statistical receiver-private oblivious transfer.

Construction. Let $(\mathsf{KGen}, \mathsf{Com}, \mathsf{H}, \mathcal{C}, \mathcal{R})$ be an SHC scheme. Let hc denote the Goldreich-Levin hardcore predicate [GL89]. The 3-round statistical receiver-private oblivious transfer proceeds as follows.

$\mathsf{OT}_1(1^\lambda)$ Execute $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda)$. Let $\mathsf{ot}_1 = \mathsf{pk}, \mathsf{st}_S = \mathsf{k}$.

$\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, \beta)$ Parse $\mathsf{ot}_1 = \mathsf{pk}$. Run $(c, \rho) \leftarrow \mathsf{Com}(\mathsf{pk}, \beta)$. Output $\mathsf{ot}_2 = c, \mathsf{st}_R = \rho$.

$\mathsf{OT}_3(1^\lambda, \mathsf{ot}_2, \mathsf{st}_S, m_0, m_1)$ Parse $\mathsf{ot}_2 = c$, and $\mathsf{st}_S = \mathsf{k}$.

For any $b \in \{0,1\}$, sample $r_b \leftarrow \{0,1\}^\lambda$, encrypt m_b as $c_b = (\mathsf{hc}(\mathsf{H}(\mathsf{k}, c, b), r_b) \oplus m_b, r_b)$.

Output $\mathsf{ot}_3 = (c_0, c_1)$.

$\mathsf{OT}_4(1^\lambda, \mathsf{ot}_1, \mathsf{ot}_3, \mathsf{st}_R)$ Parse $\mathsf{ot}_1 = \mathsf{pk}$, $\mathsf{ot}_3 = (c_0, c_1)$, and $\mathsf{st}_R = \rho$. Parse c_β as $c_\beta = (u_\beta, r_\beta)$.

Output $m' = u_\beta \oplus \mathsf{hc}(\rho, r_\beta)$.

We now prove the required properties of the protocol.

Lemma 25 (Correctness)**.** *The construction in Section 3.5 is correct.*

Proof. From the completeness of the underlying SHC scheme, we have $\Pr[\mathsf{H}(\mathsf{k}, c, \beta) = \rho] = 1$. Hence, $m' = u_\beta \oplus \mathsf{hc}(\rho, r_\beta) = \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, \beta), r_\beta) \oplus m_\beta \oplus \mathsf{hc}(\rho, r_\beta) = m_\beta$. □

Lemma 26 (Statistical Receiver-Privacy)**.** *If the underlying SHC is statistical (resp. perfect) hiding, then the construction above is statistical (resp. perfect) receiver-private.*

Proof. From the statistical hiding property of the SHC scheme, for any pk, we have $\mathsf{SD}(\mathsf{ot}_2^0, \mathsf{ot}_2^1) \leq \mathsf{neg}(\lambda)$, where $(\mathsf{ot}_2^b, \rho^b) \leftarrow \mathsf{Com}(\mathsf{pk}, b)$ for any $b \in \{0,1\}$. Hence, for any ot_1, $\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, 0)$ and $\mathsf{OT}_2(1^\lambda, \mathsf{ot}_1, 1)$ are statistically (resp. perfectly) close. □

Lemma 27 (Game-based Computational Sender-Privacy)**.** *If the underlying SHC scheme is computational binding, then the* 3*-round oblivious transfer constructed above is game-based computational sender-private.*

Proof. For any probabilistic polynomial time adversary $\mathcal{A}_0, \mathcal{A}_1$ and any probabilistic polynomial time malicious receiver $\mathcal{R}^*$ with $\mathsf{Adv}(\mathcal{A}_0, \mathcal{A}_1, \mathcal{R}^*) > \delta$, where δ is a non-negligible function of λ. Then, with probability at least $\delta/2$ over $\mathsf{View}(\mathcal{R}^*)$,

$$\begin{aligned}\exists\ \mathbf{m}_0 \in \{0,1\}^2, \mathbf{m}_1 \in \{0,1\}^2 : &\left|\Pr[\mathcal{A}_0(\mathsf{View}(\mathcal{R}^*)) \text{ wins } \mathsf{G}_0(\mathbf{m}_0)] - \frac{1}{2}\right| > \frac{\delta}{2} \wedge \\ &\left|\Pr[\mathcal{A}_1(\mathsf{View}(\mathcal{R}^*)) \text{ wins } \mathsf{G}_1(\mathbf{m}_1)] - \frac{1}{2}\right| > \frac{\delta}{2}\end{aligned}$$

Denote this fraction of $\mathsf{View}(\mathcal{R}^*)$ as GOOD. Randomly sample $\overline{\mathbf{m}}_0, \overline{\mathbf{m}}_1 \leftarrow \{0,1\}^2$. With probability $1/16$, we have $\overline{\mathbf{m}}_0 = \mathbf{m}_0 \wedge \overline{\mathbf{m}}_1 = \mathbf{m}_1$.

From Goldreich-Levin Theorem [GL89], there exists two inverters $\mathcal{A}'_0, \mathcal{A}'_1$ such that $\mathcal{A}'_0$ takes input $(\mathsf{View}(\mathcal{R}^*), r_0, \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, 1), r_1) \oplus m_1, r_1)$, output x'_0. $\mathcal{A}'_1$ takes input $(\mathsf{View}(\mathcal{R}^*), r_1, \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, 0), r_0) \oplus m_0, r_0)$, output x'_1. Furthermore, the inverters $\mathcal{A}'_0, \mathcal{A}'_1$ satisfy the property that for any $\mathsf{v} \in \mathsf{GOOD}$ and $\overline{\mathbf{m}}_0 = \mathbf{m}_0 \wedge \overline{\mathbf{m}}_1 = \mathbf{m}_1$, $\Pr[x'_0 = \mathsf{H}(\mathsf{k}, c, 0)] > \delta'$ and $\Pr[x'_1 = \mathsf{H}(\mathsf{k}, c, 1)] > \delta'$, where $\delta' = \delta'(\lambda)$ is a non-negligible function. We construct the following adversary $\mathcal{A}$ to attack the computational binding property of the SHC scheme.

Adversary $\mathcal{A}(1^\lambda, \mathsf{pk})$ Set random coins and execute $\mathcal{R}^*$. Send $\mathcal{R}^*$ the first round message $\mathsf{ot}_1 = \mathsf{pk}$, then $\mathcal{R}^*$ replies ot_2^*. Sample $r_0 \leftarrow \{0,1\}^\lambda, b_1 \leftarrow \{0,1\}, r_1 \leftarrow \{0,1\}^\lambda$, then execute $x'_0 \leftarrow \mathcal{A}'_0(\mathsf{View}(\mathcal{R}^*), r_0, b_1, r_1)$. Sample $r'_1 \leftarrow \{0,1\}^\lambda, b_0 \leftarrow \{0,1\}, r'_0 \leftarrow \{0,1\}^\lambda$, then execute $x'_1 \leftarrow \mathcal{A}'_1(\mathsf{View}(\mathcal{R}^*), r'_1, b_0, r'_0)$. Output $(c = \mathsf{ot}_2^*, x'_0, x'_1)$. We now prove that the advantage of $\mathcal{A}$ satisfies

$$\mathsf{Adv}(\mathcal{A}) = \Pr\left[(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda), (c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk}) : \substack{\rho_0 = \mathsf{H}(\mathsf{k}, c, 0) \wedge \\ \rho_1 = \mathsf{H}(\mathsf{k}, c, 1)}\right] \geq \frac{\delta \cdot \delta'^2}{128}$$

Hybrids H_0 $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda)$. Set random coins and execute $\mathcal{R}^*$. $\mathcal{R}^*$ replies ot_2^*. Sample $r_0 \leftarrow \{0,1\}^\lambda, r_1 \leftarrow \{0,1\}^\lambda$. Let $b_1 = \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, 1), r_1) \oplus m_1$. Execute $x'_0 \leftarrow \mathcal{A}'_0(\mathsf{View}(\mathcal{R}^*), r_0, b_1, r_1)$. Sample $r'_0 \leftarrow \{0,1\}^\lambda, r'_1 \leftarrow \{0,1\}^\lambda$. Let $b_0 = \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, 0), r'_0) \oplus m_0$. Execute $x'_1 \leftarrow \mathcal{A}'_1(\mathsf{View}(\mathcal{R}^*), r'_1, b_0, r'_0)$. If $\rho_0 = \mathsf{H}(\mathsf{k}, c, 0) \wedge \rho_1 = \mathsf{H}(\mathsf{k}, c, 1)$, then output 1; else output 0.

Hybrids H_1 $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda)$. Set random coins and execute $\mathcal{R}^*$. $\mathcal{R}^*$ replies ot_2^*. Sample $r_0 \leftarrow \{0,1\}^\lambda, r_1 \leftarrow \{0,1\}^\lambda$. $\underline{\text{Let } b_1 \leftarrow \{0,1\}}$. Execute $x'_0 \leftarrow \mathcal{A}'_0(\mathsf{View}(\mathcal{R}^*), r_0, b_1, r_1)$. Sample $r'_0 \leftarrow \{0,1\}^\lambda, r'_1 \leftarrow \{0,1\}^\lambda$. $\underline{\text{Let } b_0 \leftarrow \{0,1\}}$. Execute $x'_1 \leftarrow \mathcal{A}'_1(\mathsf{View}(\mathcal{R}^*), r'_1, b_0, r'_0)$. If $\rho_0 = \mathsf{H}(\mathsf{k}, c, 0) \wedge \rho_1 = \mathsf{H}(\mathsf{k}, c, 1)$, then output 1; else output 0.

Hybrids H_2 $(\mathsf{pk}, \mathsf{k}) \leftarrow \mathsf{KGen}(1^\lambda), (c, \rho_0, \rho_1) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pk})$. If $\rho_0 = \mathsf{H}(\mathsf{k}, c, 0) \wedge \rho_1 = \mathsf{H}(\mathsf{k}, c, 1)$, then output 1; else output 0.

From the construction of $\mathcal{A}$, the hybrids H_1 and H_2 are identical. Hence, $\mathsf{Adv}(\mathcal{A}) = \Pr[\mathsf{H}_2 = 1] = \Pr[\mathsf{H}_1 = 1]$. Furthermore, in hybrids H_1, with probability $1/4$, $b_1 = \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, 1), r_1) \oplus m_1 \wedge b_0 = \mathsf{hc}(\mathsf{H}(\mathsf{k}, c, 0), r'_0) \oplus m_0$. Conditioned on such event, H_0 and H_1 are identical. Hence, $\Pr[\mathsf{H}_1 = 1] \geq \Pr[\mathsf{H}_0 = 1]/4$. In hybrid H_0, the fraction of $\mathsf{View}(\mathcal{R}^*) \in \mathsf{GOOD}$ is at least $\delta/2$. With probability $1/16$, the guess of $\mathbf{m}_0, \mathbf{m}_1$ is correct. With probability δ'^2, both $\mathcal{A}'_0$ and $\mathcal{A}'_1$ inverts correctly. Hence, $\mathsf{Adv}(\mathcal{A}) \geq \frac{\delta}{2} \cdot \frac{1}{16} \cdot \delta'^2 \cdot \frac{1}{4} = \delta \cdot \delta'^2/128$. If $\delta(\lambda)$ is non-negligible, then $\mathsf{Adv}(\mathcal{A})$ is also non-negligible. This contradicts with the computational binding property of the SHC scheme. □

7 Acknowledgement

The first author is supported in part by the NSF award 1916939, a gift from Ripple, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cylab seed funding award. The second and third author are supported in part by an NSF SaTC award 1814919, DARPA Safeware award W911NF-15-C-0213, Samsung faculty award, JHU Catalyst award, and NSF CAREER award 1942789. The last author conducted part of the research while at the Simons Institute for the Theory of Computing.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer, Heidelberg, Germany.

[Bab85] László Babai. Trading group theory for randomness. In *17th ACM STOC*, pages 421–429, Providence, RI, USA, May 6–8, 1985. ACM Press.

[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, Panaji, India, November 11–14, 2018. Springer, Heidelberg, Germany.

[BFJ$^+$20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. *EUROCRYPT*, 2020.

[BGI$^+$17] Saikrishna Badrinarayanan, Sanjam Garg, Yuval Ishai, Amit Sahai, and Akshay Wadia. Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 275–303, Hong Kong, China, December 3–7, 2017. Springer, Heidelberg, Germany.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 299–315, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.

[BP15] Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

[CCH$^+$19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090, Phoenix, AZ, USA, June 23–26, 2019. ACM Press.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187, Santa Barbara, CA, USA, August 21–25, 1994. Springer, Heidelberg, Germany.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218, Dallas, TX, USA, May 23–26, 1998. ACM Press.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Heidelberg, Germany.

[DGH$^+$19] Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. *IACR Cryptology ePrint Archive*, 2019:414, 2019.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *50th FOCS*, pages 251–260, Atlanta, GA, USA, October 25–27, 2009. IEEE Computer Society Press.

[DMP88] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 52–72, Santa Barbara, CA, USA, August 16–20, 1988. Springer, Heidelberg, Germany.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st FOCS*, pages 283–293, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534, New York, NY, USA, October 17–19, 1999. IEEE Computer Society Press.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317, St. Louis, MO, USA, October 22–24, 1990. IEEE Computer Society Press.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32, Seattle, WA, USA, May 15–17, 1989. ACM Press.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229, New York City, NY, USA, May 25–27, 1987. ACM Press.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

[GOS06a] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Heidelberg, Germany.

[GOS06b] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.

[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, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

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

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 78–95, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31, Chicago, IL, USA, May 2–4, 1988. ACM Press.

[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, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In Chris Umans, editor, *58th FOCS*, pages 564–575, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press.

[LVW19] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. 2-message publicly verifiable WI from (subexponential) LWE. *IACR Cryptology ePrint Archive*, 2019:808, 2019.

[LVW20] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Statistical zapr arguments from bilinear maps. *EUROCRYPT*, 2020.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th SODA*, pages 448–457, Washington, DC, USA, January 7–9, 2001. ACM-SIAM.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 160–176, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. *Technical Report TR-81, Harvard University*, 1981.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[WW06] Stefan Wolf and Jürg Wullschleger. Oblivious transfer is symmetric. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 222–232, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167, Toronto, Ontario, Canada, October 27–29, 1986. IEEE Computer Society Press.