A Framework for Statistically Sender Private OT with Optimal Rate

Pedro Branco[1(✉)], Nico Döttling[2], and Akshayaram Srinivasan[3]

[1] Max Planck Institute for Security and Privacy, Bochum, Germany
pedrodemelobranco@gmail.com
[2] Helmholtz Center for Information Security (CISPA), Saarbrücken, Germany
[3] Tata Institute of Fundamental Research, Mumbai, India

Abstract. Statistical sender privacy (SSP) is the strongest achievable security notion for two-message oblivious transfer (OT) in the standard model, providing statistical security against malicious receivers and computational security against semi-honest senders. In this work we provide a novel construction of SSP OT from the Decisional Diffie-Hellman (DDH) and the Learning Parity with Noise (LPN) assumptions achieving (asymptotically) optimal amortized communication complexity, i.e. it achieves rate 1. Concretely, the total communication complexity for k OT instances is $2k(1 + o(1))$, which (asymptotically) approaches the information-theoretic lower bound. Previously, it was only known how to realize this primitive using heavy rate-1 FHE techniques [Brakerski et al., Gentry and Halevi TCC'19].

At the heart of our construction is a primitive called statistical co-PIR, essentially a a public key encryption scheme which statistically erases bits of the message in a few hidden locations. Our scheme achieves nearly optimal ciphertext size and provides statistical security against malicious receivers. Computational security against semi-honest senders holds under the DDH assumption.

1 Introduction

Oblivious Transfer. Oblivious transfer (OT) [46] is one of the central objects of study in secure computation: OT is complete for secure two- and multiparty computation in the sense that given a secure OT protocol, any distributed function can be computed securely. In its basic form, OT is a protocol between a receiver, holding a bit $b \in \{0,1\}$ and a sender holding two bits $m_0, m_1 \in \{0,1\}$. It allows the receiver to retrieve the bit m_b in such a way that the sender learns nothing about b, while the receiver learns nothing about m_{1-b}.

In the two party setting, OT protocols[1] cannot be information-theoretically secure against both parties, hence a cryptographic communication overhead of size $\mathsf{poly}(\lambda)$ is necessary. To amortize this overhead, one of the protocol parameters has to grow polynomially. In string-OT, the sender transfers (potentially) long message strings $\mathbf{m}_0, \mathbf{m}_1 \in \{0,1\}^{\mathsf{poly}(\lambda)}$.

[1] without the help of additional trust assumptions such as e.g. secure hardware.

H. Handschuh and A. Lysyanskaya (Eds.): CRYPTO 2023, LNCS 14081, pp. 548–576, 2023.
https://doi.org/10.1007/978-3-031-38557-5_18

The batch-setting offers a different angle on amortization: Rather than increasing the length of the messages to size $\mathsf{poly}(\lambda)$, one can bundle the joint execution of many (say, k) OT-instances into a single protocol.

Oblivious Transfer with High Rate. Minimizing the communication overhead of OT in the string and batch setting has received considerable attention in recent years [1,9–13,22,23,25,31]. This line of research has culminated in constructions of OT protocols from several hardness assumptions [18,20,26] achieving optimal communication of $2k(1+o(1))$, which approaches the information theoretic lower-bound $2k$.[2] The rate of an OT protocol in the batch setting is defined as the ratio between the information-theoretic lower bound and the overall communication of a given protocol, hence protocols with communication $2k(1 + o(1))$ have rate $1/(1 + o(1)) = 1 - o(1)$ which approaches 1.

Compared to "low-rate" OT protocols, say protocols with rate $\leq 1/2$, such high-rate OT protocols have been notoriously hard to build and there is a fundamental reason for this. Achieving communication seems to involve a *phase-transition* and should be viewed as more than a *constant factor* improvement. For example, (two-message) OT with download-rate beyond this threshold implies the existence of lossy trapdoor functions [23][3] or succinct two-round protocols for branching program evaluation and PIR schemes [23,35].

The techniques developed in this context found surprising applications, and where instrumental in several constructions of correlation-intractable hash functions, which gave rise to non-interactive zero-knowledge protocols [21,36] or batch arguments [39] from weaker assumptions.

The goal of this paper is to study what is the strongest notion of security we can achieve for OT in the plain model (i.e., without any trusted setup assumptions) while preserving optimal communication.

Statistical Sender Privacy. The standard (simulation-based) security definition of OT is given with respect to semi-honest adversaries. To achieve simulation-based security against malicious parties, one has to either rely on trusted setup assumptions (or random oracles), or increase the round complexity of the protocol. Statistical sender privacy provides a meaningful relaxation to the standard notion of malicious security and has been shown to be achievable with *two-message protocols without making use of setup assumptions* [2,43]. Since then, SSP OT was built from several assumptions such as DDH [1,2,43], QR (or DCR) [30], lattice based assumptions [19,42], or LPN together with a derandomization assumption [8].

SSP OT protocols provide security against a computationally bounded sender, but a strong statistical security notion against malicious and unbounded receivers. In essence, this notion requires the existence of an *unbounded* simulator which extracts the receiver's choice bit b^* from his (potentially malformed)

[2] The communication complexity of any, even insecure, OT is at least $2k$. We can achieve this by sacrificing sender's security (the sender sends both (m_0, m_1)) or receiver's security (the receiver sends b and obtains m_b from the sender).

[3] This is the main reason for why one could expect such a protocol to inherently be heavier on public-key operations.

protocol message, and simulates the response of an honest sender given only the message m_{b^*} the receiver is supposed to obtain.

SSP OT has been the critical ingredient in the construction of several strong primitives, most notably the construction of malicious circuit-private FHE [44], two-round statistical zaps [5,6,29,37,38], non-malleable commitments [41] or two-party computation with statistical security [7,40].

SSP OT with Rate 1. The rate-1 and SSP properties are not *orthogonal*, but intricately connected. Specifically, SSP OT can be constructed from any rate 1 OT [6,23]. The catch is, however, that this transformation (and in fact any such generic transformation) does not preserve rate 1, but necessarily makes the rate drop below 1/2.

Further note that rate 1 (batch or string) OT is by no means automatically SSP. For string OT a malicious receiver might learn half of the bits of $\mathbf{m}_0$ and $\mathbf{m}_1$, whereas for batch OT a malicious receiver might learn both message bits m_0 and m_1 for some of the batch instances. Indeed, with no safeguards against such attacks in place, it is relatively straightforward to construct malformed receiver messages to implement this attack e.g. against the scheme of Brakerski et al. [18].

One may thus raise the question whether rate 1 and SSP security can actually be achieved simultaneously. Indeed, this question was answered affirmatively by Brakerski et al. [20] and by Gentry and Halevi [26] under the LWE assumption, whose constructions of rate-1 statistically circuit private FHE give rise to rate-1 SSP OT, both in the string OT and the batch OT setting.

Their constructions, however, relies on heavy FHE machinery and on specifics of the LWE problem which allow for *ciphertext compression*. As a consequence, their result should be seen as a feasibility result. Furthermore, due to its heavy reliance on FHE-specific (non-black-box) techniques their result does not provide a recipe on how to construct high-rate SSP OT from a wider range of assumptions, or how to realize this primitive with concrete efficiency. This is unsatisfactory both from a theoretical and an applied perspective: the foundation of rate 1 SSP OT is as narrow as that of FHE, and the heavy non-black-box machinery in their construction constitutes a serious roadblock for actually using this primitive.

More recently, in the string OT setting, Aggarwal et al [1] showed that *download* rate-1 SSP string OT (for asymptotically long strings) can be achieved from the standard DDH assumption via a fully black box construction.

However, in the setting of rate-1 SSP batch OT, the FHE-based constructions of [20,26] are currently the only candidates.

1.1 Our Results

A Framework for SSP OT with Optimal Rate. Our main result is a new framework for the construction of SSP batch OT schemes with optimal communication complexity in the plain model (i.e., no random oracles or common reference string).

Statistical co-PIR. Our framework is a refinement of the blueprint of [18]. One of the key tools in [18] is a primitive called co-PIR. On a high level, a co-PIR scheme can be seen as a rate-1 public key encryption scheme (for long messages) which erases some bits of the sender's message in a way such that they are not recoverable by the receiver.

We identify the construction of co-PIR in [18] as one of the main bottlenecks towards an SSP secure rate 1 batch OT construction. The main reason is that their construction is only computationally secure, and the *lost bits* of the sender's message are only computationally hidden from the receiver.

Our first contribution is a statistically secure construction of co-PIR, which guarantees that the lost bits of the sender's message are *statistically hidden* from the receiver. As such, a statistical co-PIR scheme can be seen *somewhere statistically hiding* encryption scheme.

Theorem 1 (Informal). *There exists a co-PIR scheme that is statistically secure against malicious receivers and computationally secure against semi-honest senders assuming the DDH assumption. Additionally, the scheme fulfills the following efficiency properties:*

- *The sender's computational complexity is subquadratic in the size of the database,* $|\mathbf{D}|^{1+\varepsilon} \cdot t \cdot \mathsf{poly}(\lambda)$ *where* $\mathbf{D}$ *represents the input of the sender and* t *represents the size of the receiver's input and* $1 > \varepsilon > 0$*. The receiver's computational complexity is* $|\mathbf{D}| \cdot t \cdot \mathsf{poly}(\lambda)$*.*
- *The size of the sender's message is* $|\mathbf{D}| + o(|\mathbf{D}|)$ *when* $t = o(|\mathbf{D}|)$*.*

We further provide a generic construction of a co-PIR scheme from any (statistically sender private) rate-1 block-PIR scheme, i.e. a PIR scheme which transfers large blocks. Such rate-1 block PIR schemes can be constructed from the DDH assumption [1]. A comparison between different co-PIR schemes is presented in Table 1.

Rate-1 SSP batch OT from DDH and LPN. We provide instantiations of the primitives of our framework from the DDH assumption. We thus obtain a rate-1 SSP batch OT scheme from the DDH and LPN (with inverse polynomial noise rate) assumptions. Specifically, to execute k independent OTs, the overall communication complexity required by our protocol is $2k(1 + o(1))$.

Theorem 2 (Informal). *There exists a SSP OT scheme with optimal rate where security for the receiver holds assuming both DDH and LPN (with inverse polynomial noise-rate) assumptions.*

Our result improves upon Brakerski et al. [18] (henceforth, denoted as BBDP scheme) in terms of security: Our work achieves a stronger notion of security, namely we prove security against unbounded malicious receivers whereas the BBDP scheme achieves only computational security against semi-honest receivers. We stress that, although the BBPD scheme achieves download rate 1, it

Table 1. Comparison between existing co-PIR schemes. Here, $\mathbf{D}$ represents the input of the sender (i.e., the database), t represents the size of the receiver's input (i.e., how many indices will be erased) and $1 > \varepsilon > 0$. SSP stands for statistical sender privacy. All schemes achieve (asymptotic) download-rate 1 and the receiver's message is of size $\mathsf{poly}(t, \lambda) \cdot \mathsf{polylog}(|\mathbf{D}|)$. For all schemes, the receiver's computational complexity is $|\mathbf{D}| \cdot t \cdot \mathsf{poly}(\lambda)$.

	Security	Hardness Assumption	Sender's Work
[18]	Semi-honest	DDH, QR, LWE	$\|\mathbf{D}\| \cdot t \cdot \mathsf{poly}(\lambda)$
This work	SSP	DDH	$\|\mathbf{D}\|^{1+\varepsilon} \cdot t \cdot \mathsf{poly}(\lambda)$
This work (Full version)	SSP	Rate-1 PIR	$\|\mathbf{D}\|^2 \cdot t \cdot \mathsf{poly}(\lambda)$

only provides *computational* (instead of statistical) security against semi-honest receivers. This is because a computationally unbounded semi-honest receiver has enough information to break a subset of the OTs. A comparison with BBDP is given in Table 2

Table 2. Comparison between existing optimal-rate OT schemes. Here, k represents the number of OT executions, and $1 > \varepsilon_0, \varepsilon_1 > 0$. SSP stands for statistical sender privacy. For all schemes, the receiver's computational complexity is slightly superlinear $k^{1+\varepsilon} \cdot \mathsf{poly}(\lambda)$.

	Security	Hardness Assumption	Sender's Work
[18]	Semi-honest	$\left\{\begin{matrix}\text{DDH, QR,}\\ \text{LWE}\end{matrix}\right\}$ + LPN	$k^{1+\varepsilon_0} \cdot \mathsf{poly}(\lambda)$
This work Sect. 10	SSP	DDH + LPN	$k^{1+\varepsilon_1} \cdot \mathsf{poly}(\lambda)$

Communication-Efficient 2PC. As an application of main result, we give a construction of a 2PC protocol that has statistical security against one of the parties and has constant communication overhead. We obtain this protocol by instantiating the GMW protocol [28] using our SSP OT scheme. An informal statement of this result is given below.

Theorem 3 (Informal). *There exists a two-party secure computation scheme with communication complexity of* $\mathcal{O}(|\mathcal{C}|) + |x| + |y| + \mathsf{poly}(\lambda)$ *where* $\mathcal{C}$ *is the circuit being computed and* x, y *are the inputs of the parties. The scheme achieves statistical security against one of the parties and computational security against the other one (assuming both DDH and LPN assumptions) in the semi-honest setting.*

Previously, 2PC protocols for general circuits with constant overhead in the size of the circuit (or, better) and which provide semi-honest statistical security against one of the parties were known either from circuit-private FHE [44] or SSP OT along with PRGs in NC^0 [34].

2 Technical Overview

Throughout this technical overview, we refer to the ratio between the size of the receiver's protocol message and the size of the receiver's input as the *upload rate*, and the ratio between the size of the sender's protocol message and the size of the receiver's output as the *download rate*.

2.1 Optimal-Rate OT Secure Against Semi-honest Adversaries

Our starting point is the recent construction of *semi-honestly secure* rate-1 batch OT by Brakerski et al. [18]. In their construction semi-honest security against both senders and receivers holds computationally.

The core idea in [18] is the following: The receiver encrypts his choice-bits under a specific rate-1 private key encryption scheme, and encrypts the (short) keys under a linearly homomorphic public-key encryption (LHE) scheme. The private-key encryption scheme has the feature that approximate decryption is a linear operation. This allows the sender to decrypt the private-key ciphertext *under the hood* of the LHE an thus obtain a noisy LHE encryption of the receiver's choice. The actual OT function $f(\mathbf{x}) = (\mathbf{m}_1 - \mathbf{m_0}) \odot \mathbf{x} + \mathbf{m}_0$ (where $(\mathbf{m}_0, \mathbf{m}_1)$ is the sender's input) can now be evaluated homomorphically on the receiver's choice bits[4]. Here, $\odot$ denotes the component-wise multiplication.

Given that the LHE scheme supports *post-evaluation ciphertext compression*, the ciphertext thus generated can be compressed into a rate-1 ciphertext, which is then sent to the receiver.

In [18], the private-key scheme is instantiated from the LPN assumption and the LHE with post-evaluation compression mechanism using decisional Diffie-Hellman (DDH), quadratic residuosity (QR) or learning with errors (LWE) by adapting the ciphertext compression techniques from [15,17,18,23].

Co-private Information Retrieval. While the basic outline of the above scheme intuitively makes sense, there is a subtle issue we have glossed over: the decryption of the private key scheme is only approximate. Consequently, this will lead to a correctness error which causes the receiver to learn outputs he was not supposed to learn.

[4] For subtle technical reasons, the decryption and OT functions are combined into a single linear function.

More concretely, the private-key encryption scheme in [18] is realized as a basic LPN encryption scheme, where $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{n \times m}$ is a public random matrix, $\mathbf{s} \in \mathbb{F}_2^n$ is the secret key, and ciphertexts are of the form $\mathbf{c} = \mathbf{sA} + \mathbf{e} + \mathbf{b}$, where $\mathbf{e} \in \mathbb{F}_2^m$ is a sparse random noise term, and $\mathbf{b}$ is the vector of choice bits . The noisy plaintext can be recovered by computing $\mathbf{c} - \mathbf{sA} = \mathbf{b} + \mathbf{e}$.

Consequently, in positions i where $\mathbf{e}_i = 1$, the receiver will obtain the wrong OT output, namely $\vec{m}_{1-\vec{b}_1,i}$. Note that this does not just constitute a correctness issue, but a security issue as the receiver is not supposed to learn this value.

To address this issue, [18] introduced a new primitive called co-Private Information Retrieval (co-PIR). A co-PIR scheme allows a receiver to retrieve a database from a sender with the guarantee that some positions (unknown to the sender) are erased. More precisely, in a co-PIR scheme, the receiver starts by choosing a subset of indices $S \subset [m]$ and computes a first message copir_1. S denotes the set of indices that the receiver wants to be erased. The sender, with input a database $\mathbf{D} \in \{0,1\}^m$, computes a second message copir_2 that allows the receiver to retrieve a database $\tilde{\mathbf{D}}$. The correctness property guarantees that $\mathbf{D}$ and $\tilde{\mathbf{D}}$ coincide for all the locations $[m] \setminus S$. For security, we require that the sender obtains no information about the receiver's input and the receiver in turn learns nothing about the positions $\mathbf{D}_i$ for $i \in S$.[5] In terms of efficiency, we require that the size of the receiver's message copir_1 to only grow polylogarithmically in the size of the database m and polynomially on the size of S. Moreover, the size of sender's message copir_2 should be $\mathbf{D} + o(\mathbf{D})$ and we call such a co-PIR scheme to have near optimal download rate.

[18] provided a computationally secure construction of co-PIR from puncturable pseudorandom functions and PIR, or alternatively GGM PRFs [27] and (low-rate) OT following [10] (also known as punctured OT [16]). Moreover, these constructions achieve near optimal download rate. From a technical perspective, these constructions are *inherently* limited to computational security due to the way they use puncturable PRFs.

The above-mentioned issue can now be addressed as follows using both co-PIR and a (sender-private) PIR: The receiver generates a co-PIR message which erases all the locations corresponding to LPN errors (note that the error-locations are known to the receiver). Furthermore, he generates PIR instances which retrieve information at the locations corresponding to LPN errors. The sender will now transmit the compressed LHE ciphertext using the co-PIR, and use the PIR scheme to transmit the correct outputs at the erased positions.

2.2 Towards Statistical Sender Privacy

In order to adapt the BBDP-framework to the setting of statistical sender privacy *while preserving rate-1*, we encounter the following challenges.

[5] Co-PIR can be seen as the opposite of PIR: In a PIR the receiver retrieves the positions of the database that it is asking for, whereas in a co-PIR it gets the entire database except for those positions.

1. **Statistical co-PIR.** Clearly, the biggest issue with the BBDP construction with respect to SSP security lies in the fact that their co-PIR scheme offers only computational security, *even against semi-honest receiver*. Hence it seems inevitable that we have to take a different route to construct statistical co-PIR. Additionally, we need this statistical secure co-PIR scheme to have near optimal download rate.
2. **Consistency of Inputs.** We need the input set S sent by the receiver as part of PIR and co-PIR messages to be the same. Otherwise, a malicious receiver can cheat and learn both messages $m_{0,i}$ and $m_{1,i}$ for some position i and thus, breaking the sender security of OT.
3. **Well-formedness of ciphertexts.** The protocol described above assumes that the ciphertext ct (encrypting the LPN secret) generated by the receiver is well-formed. Namely, this ciphertext should encrypt bits and have a special structure that allows for packing of ciphertexts.

In the following, we will outline our approach to deal with these issues.

2.3 Statistical Co-PIR

Our main challenge is to construct a co-PIR scheme that provides statistical security against malicious receivers and has near optimal download rate. We build such a co-PIR scheme in a sequence of steps:

1. We start by building a one-query statistical semi-honest co-PIR, which erases only a single block of bits and provides semi-honest security for both the receiver and the sender.
2. We then show how to achieve a one-query statistical semi-honest co-PIR that erases a single bit (instead of an entire block).
3. In the next transformation, we show how to bootstrap a co-PIR that only allows to erase one bit into one where multiple bits are erased.
4. Finally, we show how to achieve statistical sender privacy.

One-Query Statistical Semi-honest Co-PIR. We first tackle the simpler task of constructing a statistical co-PIR which only erases one position of the database and the security is required to hold only against semi-honest adversaries. We will call this primitive a one-query semi-honest co-PIR.

One-Query Semi-honest co-PIR from PIR. One-query semi-honest co-PIR can be constructed in a generic way from PIR. The receiver's input to the PIR corresponds to the position that it wants to be erased. The sender's input to the PIR corresponds to vectors $\hat{\mathbf{D}}_1, \ldots, \hat{\mathbf{D}}_m$ where each $\hat{\mathbf{D}}_i$ corresponds to the database $\mathbf{D}$ with the i-th position erased. By the correctness of the PIR, the receiver obtains $\hat{\mathbf{D}}_i$.

For the resulting co-PIR scheme to be rate-1 and provide statistical security for the sender, we need that the underlying PIR to fulfill these requirements. Such a PIR scheme was recently constructed in the work of Aggarwal et al. [1].

However, a drawback of this construction is that the sender's work is proportional to $|\mathbf{D}|^2$, which is the size of the sender's input to the PIR, whereas the receiver's work is proportional to $|\mathbf{D}|$. We now explain how to build a co-PIR scheme which achieves better efficiency for the sender.

All-But-One Lossiness. Our first observation is that a one-query statistical co-PIR resembles a primitive called *all-but-one trapdoor lossy function* (ABO-TDF) [45]. Loosely speaking, an ABO-TDF is a function parametrized by some public key and which is invertible everywhere except for some specified one branch where it loses information. Crucially, the public key should not reveal about the lossy branch.

Peikert and Waters provide a simple construction of ABO-TDF from a linear homomorphic scheme LHE such as El Gamal: to generate an ABO-TDF public key which is lossy on a branch i^*, one first generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LHE.KeyGen}(1^\lambda)$ and encrypts $\mathsf{ct} \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, i^*)$. The new ABO-TDF public key is composed by $(\mathsf{pk}, \mathsf{ct})$. To encrypt a message $m \in \mathbb{Z}_2$ under index i, we homomorphically compute the function $f(x) = (i - x) \cdot m$ and obtain a new ciphertext $\tilde{\mathsf{ct}}$.

It is easy to see that for all $i \neq i^*$, decryption can recover $m = \mathsf{LHE.Dec}(\mathsf{sk}, \tilde{\mathsf{ct}}) \cdot (i - i^*)^{-1}$. However, when $i = i^*$ all information about m is statistically hidden, assuming that LHE is function private.

A Simple Statistical co-PIR with Large Computation. In the following, let p be the order of a DDH group and LHE be a function private LHE scheme over a smaller field $\mathbb{Z}_q$ for $q = \mathsf{poly}(\lambda)$, such as the one presented in [18]. Let $\mathbf{D} \in \mathbb{Z}_q^m$ be the database of the sender, where q will be later defined. As a first approach consider the following protocol for co-PIR:

- The receiver creates a pair of public/secret keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LHE.KeyGen}(1^\lambda)$ and encrypts $\mathsf{ct} \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, i^*)$ for $i^* \in [m]$.
- For all $i \in [m]$ the sender homomorphically computes $f_i(x) = (i - x) \cdot \mathbf{D}_i$ over ct and obtains ciphertexts $\tilde{\mathsf{ct}}_1, \ldots, \tilde{\mathsf{ct}}_m$.
- For all $i \neq i^*$ the receiver obtains $\mathbf{D}_i \leftarrow \mathsf{LHE.Dec}(\mathsf{sk}, \tilde{\mathsf{ct}}_i)$.

It is easy to see that correctness holds for all $i \neq i^*$. Semi-honest security for the receiver follows from the IND-CPA of LHE and semi-honest statistical security for the sender follows from the statistical function privacy of LHE and from the fact that $(i^* - i^*) \cdot m = 0 \cdot m = 0$.

In terms of efficiency, the receiver's message is composed by a public key and an encryption and hence its size is independent of $|\mathbf{D}|$. However, the sender's message is composed by m uncompressed ciphertexts. So the scheme does not achieve near optimal download rate.

To achieve near optimal download rate, we will use the ciphertext compression technique for El Gamal presented in [18] (which is itself based on previous works [15,17,23]). These techniques are specially designed for packed El Gamal and to use these packing techniques, we need the following two conditions to hold.

1. The receiver's message needs to encrypt a matrix rather than a single value i^*, in order for packing to be possible. That is, $\mathsf{ct} \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, i^* \cdot \mathbf{I})$ where $\mathbf{I}$ is the identity matrix of size k.
2. We need $\mathbf{D}_i$ to be in $\mathbb{Z}_q^k$ for large enough k in order to amortize the size of the ciphertext for a single block. Moreover, we need that $q > m$. The latter condition comes from the fact that the operation $(i - i^*)$ needs to be performed over a modulus greater than m. If that was not the case, then it might happen that $(i - i^*) = 0 \mod q$ for $i \neq i^*$ over the integers and we will loose correctness.

This gives us a statistical semi-honest one-query co-PIR for databases of size $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m)$ where each $\mathbf{D}_i \in \mathbb{Z}_q^k$ for $q > m$.

In terms of computation, the scheme still incurs a quadratic blowup for both the sender and the receiver. All ciphertext compression mechanisms for DDH [15,17,18,23] have computational complexity scaling with q for both the sender and the receiver. Since $q > m$, for each block, both parties have to spend computational work proportional to m. Since there are m blocks, we end up with computational complexity proportional to at least m^2. Thus, we have not achieved any significant gains over the simple solution from PIR.

An Efficient Statistical co-PIR. To improve the computational complexity of the protocol, we need a way to make the complexity of encrypting and decrypting each block independent of m. Towards this goal, we use a standard trick of embedding the underlying messages in an extension field.

To be a bit more specific, our idea is to parse the database $\mathbf{D}$ as a vector over an extension field $\mathbb{F}_{2^\mu}$ where $\mu = \lceil \log m \rceil$. That is, we parse $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m) \in \mathbb{F}_{2^\mu}^{k \cdot m}$ where each $\mathbf{D}_i \in \mathbb{F}_{2^\mu}^k$.

We rely on the fact that for any two elements $\hat{x}, \hat{a} \in \mathbb{F}_{2^\mu}$, where $\hat{x} = x_1 + x_2\alpha + \cdots + x_\mu\alpha^{\mu-1}$ for some symbol α, each coefficient of the product $\hat{x} \cdot \hat{a}$ can be expressed as a linear function depending only on $\hat{a}$. That is,

$$\hat{x} \cdot \hat{a} = f_{1,\hat{a}}(\mathbf{x}) + f_{2,\hat{a}}(\mathbf{x})\alpha + \cdots + f_{\mu,\hat{a}}(\mathbf{x})\alpha^{\mu-1}$$

where $\mathbf{x} = (x_1, \ldots, x_\mu)$ and each $f_{i,\hat{a}}$ is a $\mathbb{Z}_2$-linear function that depends solely on $\hat{a}$.

Given this, the new scheme with improved computational complexity can be obtained as follows:

- Given an index $i^* \in [m]$, the receiver first decomposes it into its binary decomposition $\mathbf{i}^* = (i_1^*, \ldots, i_\mu^*)$. Then, it creates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LHE.KeyGen}(1^\lambda)$ and encrypts each i_1^*, that is, $\mathsf{ct}_j \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, i_j^* \cdot \mathbf{I})$ where $\mathbf{I}$ is the identity matrix of size k. It sends pk and the ciphertexts ct_j to the sender.
- The sender parses $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m) \in \mathbb{F}_{2^\mu}^{k \cdot m}$ where each $\mathbf{D}_i \in \mathbb{F}_{2^\mu}^k$. For each $\ell \in [m]$ it evaluates the function $f_i(\hat{\mathbf{X}}) = (\hat{\ell} \cdot \mathbf{I} - \hat{\mathbf{X}}) \cdot \mathbf{D}_\ell$ over $\mathbb{F}_{2^\mu}$ where $\hat{\ell}$ is the embbeding of ℓ in $\mathbb{F}_{2^\mu}$ (by first converting into its binary decomposition and then interpreting it as a $\mathbb{F}_{2^\mu}$ element). As we have seen, f_i can be expressed as a $\mathbb{Z}_2$-linear function. Let $\tilde{\mathsf{ct}}_\ell$ be the resulting ciphertexts. It compresses the ciphertexts $\tilde{\mathsf{ct}}_\ell$ to make them rate 1.

– Finally, the receiver decrypts each $\tilde{\mathsf{ct}}_\ell$ for $\ell \neq i^*$, interprets the result as a $\mathbb{F}_{2^\mu}^k$ vector $\mathbf{u}$ and computes $\tilde{\mathbf{D}}_\ell = (\hat{\ell} - \hat{i}^*)^{-1} \cdot \mathbf{u}$ over $\mathbb{F}_{2^\mu}$.

Correctness, semi-honest security for the receiver and semi-honest statistical sender security hold as in the protocol above.

Computational Complexity. Unlike the previous protocol, the sender needs to compress m ciphertexts of size $\mu \cdot k$. However, now all ciphertexts encrypt bits instead of messages over a larger modulus. Hence, the sender's computational complexity grows only linearly with the size $m \cdot k \cdot \mu$ of the database. Similarly, the receiver needs to decrypt the ciphertexts (encrypting bits) sent by the sender, hence the computational complexity for the receiver grows only linearly with the size of the database

Full-Fledged Statistical Co-PIR. Until now we have discussed how to obtain a semi-honest statistical one-query co-PIR. However, for our OT application, the co-PIR needs to i) provide statistical security against a malicious receiver, and ii) support more than one query. Additionally, to obtain a bit-OT, we need our co-PIR to erase a single bit of the database whereas the construction presented above only works for large erased blocks.

Bit co-PIR from Block co-PIR. We start by the simplest task of turning a co-PIR which erases an entire block, or block co-PIR, into one that erases a single bit, or bit co-PIR.

Assume that the receiver wants to erase the j^* bit of the i^* block. We show that a bit co-PIR can be built from a block co-PIR by additionally assuming the existence of a PIR scheme. The block co-PIR will erase an entire block $\mathbf{D}_{i^*} = (D_{i^*,1}, \ldots, D_{i^*,k})$. The remaining positions $D_{i^*,j}$ for $j \neq j^*$ can be sent to the receiver via a PIR. The resulting scheme can be seen as a (one-query) bit co-PIR as only D_{i^*,j^*} is erased from the perspective of the receiver. Importantly, we show that this scheme preserves security even against malicious receivers. This is because if the PIR message points to another block, then the malicious receiver obtains strictly lesser information and this does not violate the privacy of the honest sender.

Additionally, in terms of communication the scheme preserves i) short message from the receiver as the PIR receiver's message is small compared to the size of the database $|\mathbf{D}| = mk$, and ii) near optimal download rate as the sender's PIR message only grows with k.

In terms of computation, the scheme preserves the computational complexity for the receiver. However, the sender now as to run k PIR second message which makes its work to grow proportionally with $|\mathbf{D}| \cdot k$. Setting k to be sublinear in m yields subquadratic work in the size of the database for the sender.

Multiple-Query co-PIR via Recursion. We now discuss how to obtain a co-PIR where the receiver's input is a set S of indices instead of a single index.

Assume that the message of the sender can be decomposed into bit ciphertexts. That is, copir_2 can be decomposed into $(h, \alpha_1, \ldots, \alpha_m)$ where h is a header of size $\mathsf{poly}(\lambda)$ and each α_i decrypts to $D_i \in \{0,1\}$, where $\mathbf{D} = (D_1, \ldots, D_m)$ is the sender's input.

The crucial idea of this transformation is that, since the underlying one-query co-PIR has near optimal download rate, the sender can recurse the co-PIR without any blowup in the communication. Concretely the protocol works as follows:

- The receiver sends t first one-query co-PIR messages $\{\mathsf{copir}_{1,i}\}_{i\in[t]}$ to the sender, each one encoding an index a_i to be erased.
- The sender computes the first one-query co-PIR message $\mathsf{copir}_{2,1}$ using the input database $\mathbf{D} \in \{0,1\}^m$ and $\mathsf{copir}_{1,1}$. Recall that $\mathsf{copir}_{2,1}$ can be decomposed into $(h_1, \alpha_1^{(1)}, \ldots, \alpha_m^{(1)})$. The sender now creates a second $\mathsf{copir}_{2,2}$ using a new database $\mathbf{D}_1 = (\alpha_1^{(1)}, \ldots, \alpha_m^{(1)})s$ and $\mathsf{copir}_{1,2}$. The sender repeats this process until it obtains $\mathsf{copir}_{2,t}$ (together with all previous headers) and sends this to the receiver.
- The receiver can recursively decrypt each $\mathsf{copir}_{2,t+1-i}$ for $i \in [t]$. At each step, the a_{t+1-i} position of $(\alpha_1^{(t+1-i)}, \ldots, \alpha_m^{(t+1-i)})$ is erased and information about $D_{a_{t+1-i}}$ is statistically erased.

Since the underlying one-query co-PIR has near optimal download rate, each iteration of the recursion maintains this property as long as t is sublinear in the size of the initial database $\mathbf{D}$. Furthermore, if the starting co-PIR is statistically secure against malicious receivers, then so is the transformed co-PIR.

Achieving Statistical Sender Privacy Against Malicious Receivers. So far we have only discussed how to achieve semi-honest statistical security. It remains to show how to turn the protocol statistically secure for the sender against malicious receivers who might send malformed first round messages.

If we are able to guarantee that the receiver's message is well-formed, then we can use the semi-honest (statistical) security to argue malicious (statistical) security. As a first approach, we will discuss how to use a statistically sender secure *conditional disclosure of secrets* (CDS) to achieve the stronger notion of security.

Let $\mathcal{L}$ be an NP language. Recall that in a CDS scheme, the receiver holding a witness w for a statement x, sends a first message cds_1 to the sender that commits to w. The sender holding a message m computes a second CDS message cds_2 which allows the receiver to retrieve m iff $x \in \mathcal{L}$ and w is a valid witness. In terms of security, we want that if $x \notin \mathcal{L}$, then m is statistically hidden from the receiver.

It is well-known that statistically sender secure CDS schemes for NC1 can be constructed using (low-rate) SSP OT and information theoretic garbled circuits [3,4,32]. Moreover, any NP language can be verified by a NC1 circuit [24].

In order to achieve statistical sender security against malicious receivers, we will use a CDS that guarantees that the receiver's message is well-formed.

Consider the following language

$$\mathcal{L}_{\mathsf{CoPIR}'} = \{\mathsf{copir}_1 : \exists(S, r) \text{ s.t. } \mathsf{copir}_1 \leftarrow \mathsf{CoPIR}'.\mathsf{Query}(S; r)\}$$

that is the language of well formed receiver's messages, where CoPIR' is the semi-honest protocol[6] described in the previous sections. Additionally, recall that our co-PIR scheme has a decomposability feature that the sender's message copir_2 can be decomposed into $(\alpha_1, \ldots, \alpha_m)$ where each α_i encodes $\mathbf{D}_i$.

The statistically secure protocol can be roughly described as follows:

- The receiver sends a co-PIR message $\mathsf{copir}_1 \leftarrow \mathsf{CoPIR}'.\mathsf{Query}(S; r)$ for a set of indices S of size t using random coins r. It additionally sends a first message cds_1 for language $\mathcal{L}_{\mathsf{CoPIR}'}$ using (S, r) as the witness.
- The sender computes $\mathsf{copir}_2 \leftarrow \mathsf{CoPIR}'.\mathsf{Send}(\mathsf{copir}_1, \mathbf{D})$ and decomposes copir_2 into $(\alpha_1, \ldots, \alpha_m)$. It now samples random $\beta_1, \ldots, \beta_t$ and computes

  $$\mathbf{v} = (\alpha_1, \ldots, \alpha_m) + (\beta_1, \ldots, \beta_t, 0, \ldots, 0)$$

  that is the first t coordinates of copir_2 are hidden using $\beta_1, \ldots, \beta_t$. It now sends a CDS message $\mathsf{cds}_2 \leftarrow \mathsf{CDS}.\mathsf{Send}(\mathsf{cds}_1, (\beta_1, \ldots, \beta_t))$ encrypting the values $(\beta_1, \ldots, \beta_t)$.

If copir_1 is well-formed then the receiver can retrieve the values $\beta_1, \ldots, \beta_t$, recover copir_2 and retrieve $(\alpha_1, \ldots, \alpha_m)$. In this case, we can use the semi-honest (statistical) security of the underlying CoPIR' to argue that the scheme is statistically secure. On the other hand, if copir_1 is malformed, then the values $\beta_1, \ldots, \beta_t$ are statistically hidden from the receiver given that CDS is statistically secure. In this case, the values $\alpha_1, \ldots, \alpha_t$ are statistically hidden from the receiver's point of view and thus, the first t positions of $\mathbf{D}$ are statistically hidden.

In terms of communication, the scheme has near optimal download rate as the CDS communication only depends on t (i.e., the size of the receiver's message) and this is typically set to be sub-linear in the size of the database.

While this gives us a generic solution to achieve SSP co-PIR, it incurs in a huge overhead as we need to make non black-box use of the underlying semi-honest co-PIR.

A Black-Box Solution. To achieve a better concrete efficiency, we show how to build a black-box CDS scheme specifically for our purposes.

Recall that the receiver's message is composed by a ciphertext encrypting a square matrix of size k.[7] That is, a well formed receiver's message consists of $\mathsf{ct} \leftarrow \mathsf{LHE}.\mathsf{Enc}(\mathsf{pk}, b \cdot \mathbf{I})$ where $b \in \{0, 1\}$ and $\mathbf{I}$ is the identity matrix. However, if the receiver behaves maliciously, then it can encrypt any matrix $\mathbf{A}$ so that it learns partial (or total) information about the erased block.

[6] Technically speaking, we need the co-PIR scheme to be semi-malicious secure but for the sake of this overview, we will ignore this difference. Our co-PIR scheme constructed before satisfies semi-malicious security.

[7] The receiver's message is actually composed by several of these ciphertexts but for simplicity we assume that we only have one ciphertext.

Algebraic Restriction Codes. This is where algebraic restriction (AR) [1] codes come into play. Roughly speaking, an AR code restricts the class of functions that an adversary can apply over an encoded value.

More precisely, let $\hat{\mathbf{y}}_i \leftarrow \mathsf{AR.Encode}(\mathbf{y}_i)$ for $i = 1, 2$. The work of [1] provides a construction of AR codes that restrict the class of any linear function $g(\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2) = \hat{\mathbf{y}}_1 \cdot \mathbf{A} + \hat{\mathbf{y}}_2$ over the encoded values to the class of $f(\mathbf{y}_1, \mathbf{y}_2) = \mathbf{y}_1 \cdot (c \cdot \mathbf{I}) + \mathbf{y}_2$ where $\mathbf{I}$ is the identity matrix and $c \in \mathbb{Z}_p$. The security of AR codes allow to statistically simulate the evaluation of g over two encoded values $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2$ given just the output of the decoding $f(\mathbf{y}_1, \mathbf{y}_2)$ where f is a function that depends on g.

A CDS from AR Codes. Our main idea is to recast the construction of [1] in terms of CDS. Specifically, the sender sets

$$\mathbf{y}_1 = \begin{pmatrix} \mathbf{r}_0 \\ \mathbf{r}_1 \end{pmatrix} \text{ and } \mathbf{y}_1 = \begin{pmatrix} \mathbf{m} \\ \mathbf{m} - \mathbf{r}_1 \end{pmatrix}$$

where $\mathbf{r}_0, \mathbf{r}_1 \leftarrow_{\$} \mathbb{Z}_p^k$ and $\mathbf{m}$ is the message that the sender wants to send.

However, instead of evaluating $F(\mathbf{X}) = \mathbf{y}_1 \cdot \mathbf{X} + \mathbf{y}_2$ (that depends on $\mathbf{y}_1, \mathbf{y}_2$) over ct, it first AR encodes both $\mathbf{y}_1, \mathbf{y}_2$, applies $G(\mathbf{X}) = \hat{\mathbf{y}}_1 \cdot \mathbf{X} + \hat{\mathbf{y}}_2$ and finally homomorphically decodes the result.[8]

The AR codes security guarantees that the receiver will decrypt to something of the form

$$\mathbf{y}_1 \cdot (c\mathbf{I}) + \mathbf{y}_2 = \begin{pmatrix} c\mathbf{r}_0 + \mathbf{m} \\ c\mathbf{r}_1 + \mathbf{m} - \mathbf{r}_1 \end{pmatrix}.$$

If $c = 0, 1$ then the receiver can retrieve $\mathbf{m}$. However, if $c \neq 0, 1$ then the message $\mathbf{m}$ is statistically hidden from the receiver.

2.4 Consistency of Inputs

Recall that we need a mechanism to ensure that the same set is used by a malicious receiver to generate a PIR and a co-PIR message. First, note that the underlying PIR also needs to be statistically sender secure, and this can be instantiated using the scheme of [1]. Our second crucial observation is that the co-PIR receiver message comprises of a public key pk and encryptions of $a_i \cdot \mathbf{I}$ for $a_i \in \{0, 1\}$ which means that the co-PIR messages are identical to the PIR messages of the scheme of [1]![9] This means that the receiver does not have to send separate PIR messages: the sender can just interpret the co-PIR messages as PIR messages and this guarantees consistency of inputs.

[8] For this to work, we need the decoding function of the AR codes to be a linear function and this is indeed the case for AR codes from [1].

[9] Our actual co-PIR scheme is a bit more complex as it also contains PIR messages as a result of the block-to-bit transformation. However, our co-PIR scheme is still "PIR-compatible" for a variant of the PIR scheme of [1].

2.5 Well-Formedness of Ciphertexts

The last missing piece is how to ensure that the ciphertexts encrypting the LPN secrets are well-formed. These ciphertexts need to have a special structure i.e., they need to be encrypting bits, but in the current form, there is nothing that prevents the adversary from sending malformed ciphertexts and learn additional information.

Unfortunately, we cannot use a generic CDS protocol as we will lose optimal sender's message length or statistical security against the receivers. This is where we use rate-1 CDS. A rate-1 CDS is a standard statistical sender secure CDS with one additional efficiency property: we require the size of the sender's message to be $|m| + o(|m|)$ for sufficiently long m (which is larger than the size of the NP verification).

Assume for now that we have a download rate-1 CDS scheme which is statistically secure against malicious receivers. The sender encrypts its OT message using this CDS, and this message will be released to the receiver iff the ciphertexts are well-formed. Since the CDS scheme is rate-1, there is no blowup in the size of the sender's message. Moreover, the receiver's CDS message is small as its size only depends on the size of the LPN secrets and the size of the NP relation to be verified is independent of the size of the sender's input.

To construct such a rate-1 CDS scheme, we plug the (download rate-1) OT scheme of [1] together with the encryption scheme of [35] which yields a CDS scheme for branching programs (which contains NC1 circuits) and this is sufficient for our purposes.

2.6 Future Directions

Except for the use of the general purpose rate-1 CDS scheme used in the last step, our scheme uses only black-box techniques in the sense that it does not use explicit circuit-level description of cryptographic primitives. Coming up with a black-box technique that guarantees well-formedness of the ciphertexts of the receiver is a interesting open problem.

3 Preliminaries

Due to space constraints the preliminaries are presented in the full version of the paper. Here we just present non standard notation.

Let $\mathsf{Diag}(n, \mathbf{v})$ be the algorithm that takes a vector $\mathbf{v} = (v_1, \ldots, v_n) \in \{0,1\}^n$ and outputs a matrix

$$\mathbf{D} = \begin{pmatrix} v_1 & & 0 \\ & \ddots & \\ 0 & & v_n \end{pmatrix} \in \{0,1\}^{n \times n},$$

i.e. $\mathbf{D} \in \{0,1\}^{n \times n}$ is a diagonal matrix with the components of $\mathbf{v}$ on its diagonal.

Addtionally let $\mathsf{SingleRowMatrix}(\ell, n, i, \mathbf{v})$ be the algorithm that takes $i \in [\ell]$ and a row-vector $\mathbf{v} \in \{0,1\}^n$ and outputs a matrix

$$\mathbf{V} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \\ — & \mathbf{v} & — \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix} \in \{0,1\}^{\ell \times n},$$

i.e. the i-th row of $\mathbf{V}$ is $\mathbf{v}$, but $\mathbf{V}$ is 0 everywhere else.

4 Definition of Co-Private Information Retrieval

We start by defining co-PIR in an identical way as in [18]. A co-PIR is a primitive that allows a sender to transmit a database to a receiver, except for some positions which will be (statistically) erased.

Definition 4 (Co-PIR). *Let $\mathbb{H}$ be a group. A co-private information retrieval (co-PIR) scheme is parametrized by a integer $m = \mathsf{poly}(\lambda)$ and is given by a tuple of algorithms $(\mathsf{Query}, \mathsf{Send}, \mathsf{Rec})$ with the following sintax:*

- $\mathsf{Query}(1^\lambda, S)$ *takes as input the security parameter λ and a subset of indices $S = \{i_1, \dots, i_t\} \subseteq [m]$ of size t. It outputs a first co-PIR message copir_1 and a private state st.*
- $\mathsf{Send}(\mathsf{copir}_1, \mathbf{D})$ *takes as input a first co-PIR message copir_1 and a database $\mathbf{D} \in \mathbb{H}^m$.*[10] *It outputs a second co-PIR message copir_2.*
- $\mathsf{Rec}(\mathsf{copir}_2, \mathsf{st})$ *takes as input a second co-PIR message copir_2 and a private state st. It outputs a database $\tilde{\mathbf{D}} \in \mathbb{H}^m$.*

A co-PIR scheme should fulfill the following properties.

Definition 5 (Correctness). *A co-PIR scheme is said to be correct if for any $m = \mathsf{poly}(\lambda)$ and $S \subseteq [m]$*

$$\Pr\left[\mathbf{D}_{\bar{S}} = \tilde{\mathbf{D}}_{\bar{S}} : \begin{array}{c} (\mathsf{copir}_1, \mathsf{st}) \leftarrow \mathsf{Query}(1^\lambda, S) \\ \mathsf{copir}_2 \leftarrow \mathsf{Send}(\mathsf{copir}_1, \mathbf{D}) \\ \tilde{\mathbf{D}} \leftarrow \mathsf{Rec}(\mathsf{copir}_2, \mathsf{st}) \end{array}\right] = 1$$

where $\bar{S} = [m] \setminus S$. In other words, $\mathbf{D}_i = \tilde{\mathbf{D}}_i$ for all $i \notin S$.

We also define a slightly stronger notion of security that we call locally correct.

[10] We use the term bit co-PIR to denote the case when $\mathbb{H} = \{0,1\}$. Otherwise, we use the term block co-PIR.

Definition 6 (Locally correctness). *A co-PIR scheme is locally correct if the following holds: i)* copir_2 *is of the form* $(\alpha_1, \ldots, \alpha_m)$*, and ii) The* Rec *algorithm can be divided into subalgorithms* Rec_i *such that* $\mathbf{D}_i \leftarrow \mathsf{Rec}_i(\alpha_i, \mathsf{st})$ *for all* $i \notin S$*.*

Definition 7 (Efficiency). *A co-PIR scheme is said to be efficient if the it fulfills the following requirements:*

- $|\mathsf{copir}_1| = \mathsf{polylog}(|\mathbf{D}|) \cdot \mathsf{poly}(\lambda, |S|)$.[11]
- ***Download rate 1:*** *If* t *is sublinear in* $|\mathbf{D}|$ *(that is* $t = o(|\mathbf{D}|)$*) then*

$$\lim_{\lambda \to \infty} \sup \frac{|\mathsf{copir}_2|}{|\mathbf{D}|} \to 1$$

for sufficiently large $|\mathbf{D}|$ *where* $\mathbf{D} \in \mathbb{H}^m$ *and* $\mathsf{copir}_2 \leftarrow \mathsf{Send}(\mathsf{copir}_1, \mathbf{D})$.[12]

Definition 8 (Receiver security). *A co-PIR scheme* CoPIR *is said to be receiver secure if for all* $m = \mathsf{poly}(\lambda)$*, any subsets* $S_1, S_2 \subseteq [m]$ *we have that for any adversary* $\mathcal{A}$

$$\left| \begin{array}{l} \Pr\left[1 \leftarrow \mathcal{A}(k, \mathsf{copir}_1) : (\mathsf{copir}_1, \mathsf{st}) \leftarrow \mathsf{Query}(1^\lambda, S_1)\right] - \\ \Pr\left[1 \leftarrow \mathcal{A}(k, \mathsf{copir}_1) : (\mathsf{copir}_1, \mathsf{st}) \leftarrow \mathsf{Query}(1^\lambda, S_2)\right] \end{array} \right| \leq \mathsf{negl}(\lambda).$$

Sender Security. We define two notions for statistical sender security. The first one, which is the strongest one, considers malicious and computationally unbounded receivers.

Definition 9 (Statistical sender security). *A co-PIR is said to be statistically sender secure if there is a (possibly computationally inefficient) extractor* $\mathsf{CoPIR.Ext}$ *such that for any message* copir_1 *and any pair of databases* $(\mathbf{D}, \mathbf{D}')$

$$\mathsf{Send}(\mathsf{copir}_1, \mathbf{D}) \approx_s \mathsf{Send}(\mathsf{copir}_1, \mathbf{D}')$$

where $S \leftarrow \mathsf{CoPIR.Ext}(\mathsf{copir}_1)$ *and* $\mathbf{D}'_i = \mathbf{D}_i$ *for* $i \notin S$*. Here,* S *is a set of size at most* t*.*

We also consider a relaxation of sender security that only considers semi-honest and computationally unbounded receivers.

[11] We usually consider co-PIR protocols where the first message depends polylogarithmically on the size of $\mathbf{D}$, similarly to PIR protocols. However, for our OT application in Sect. 10, it is enough to consider copir_1 to depend sublinearly on the size of $\mathbf{D}$.

[12] In a co-PIR, we allow the sender's message to be of the same size of the sender's input (or even slightly larger by an additive term depending on t) instead of the usual rate-1 definition which compares the sender's message with the receiver's input. This is the reason why we e define a co-PIR to be rate-1 only for $t = o(\mathbf{D})$ erased positions, which is enough for our applications.

Definition 10 (Semi-honest statistical sender security). *A one-query co-PIR scheme* CoPIR *is said to be semi-honest sender secure if for all* $S \subseteq [m]$ *of size* t *we have that*

$$\mathsf{Send}(\mathsf{copir}_1, \mathbf{D}) \approx_s \mathsf{Send}(\mathsf{copir}_1, \mathbf{D}')$$

for all $\mathbf{D}, \mathbf{D}' \in \mathbb{H}^m$ *such that* $\mathbf{D}'_i = \mathbf{D}_i$ *for* $i \notin S$*, and all* $\mathsf{copir}_1 \leftarrow \mathsf{Query}(1^\lambda, S)$.

A co-PIR fulfilling the latter security definition is called a semi-honest co-PIR.

One-Query co-PIR. We also define a one-query co-PIR scheme, that is, a co-PIR where the receiver's query is composed by a single index.

Definition 11 (One-query co-PIR). *A one-query co-PIR scheme is identical to a co-PIR except that the input of the receiver is composed by a single index. That is, the set* S *in Definition 4 is of the form* $S = \{i^*\}$*. Correctness, statistical sender security and receiver security are defined in an analogous way.*

Self-reducibility. Another property that we will need is self-reducibility.[13] This ensures that the output of a one-query co-PIR has the same form as the database and thus can be input into a new one-query co-PIR.

Definition 12 (Self-reducibility). *A one-query co-PIR scheme is said to be self-reducible if the sender's message is of the form* $\mathsf{copir}_2 = (\mathsf{head}, \alpha_1, \ldots, \alpha_m)$*. Moreover, for any* $i^* \in [m]$*, any two databases* $\mathbf{D}, \mathbf{D}'$ *such that* $\mathbf{D}_i = \mathbf{D}'_i$ *for all* $i \neq i^*$ *and any* copir_1 *message we have that*

$$(\mathsf{head}, \alpha_1, \ldots, \alpha_{i^*-1}, \alpha_{i^*+1}, \ldots, \alpha_m) \approx_s (\mathsf{head}', \alpha'_1, \ldots, \alpha'_{i^*-1}, \alpha'_{i^*+1}, \ldots, \alpha'_m)$$

where $\mathsf{copir}_2 = (\mathsf{head}, \alpha_1, \ldots, \alpha_m) \leftarrow \mathsf{Send}(\mathsf{copir}_1, \mathbf{D})$, $\mathsf{copir}'_2 = (\mathsf{head}', \alpha'_1, \ldots, \alpha'_m) \leftarrow \mathsf{Send}(\mathsf{copir}_1, \mathbf{D}')$.

In other words, self-reducibility states that all information about the block/bit D_i of the database is contained in a single block/bit of the copir_2 message. This property will be essential for recursion.[14]

[13] The work of [14] defines an identical property for OT.

[14] In this definition we assume that the i-th block/bit of copir erases D_i. However, this does not need to be the case in general: it might happen that the i-th block/bit of copir erases D_j, which is what happens with our construction in Sect. 6. However, both definitions are equivalent up to a reordering of the database.

PIR Compatibility. Let PIR be a PIR scheme and CoPIR be a co-PIR scheme. We say that CoPIR is PIR-compatible if the first message $\mathsf{copir}_1 \leftarrow \mathsf{CoPIR.Query}(1^\lambda, S)$ can be used as a first message of the PIR scheme. That is, we can parse copir_1 as q and compute $\mathsf{r} \leftarrow \mathsf{PIR.Send}(\mathbf{D}, \mathsf{q})$ while preserving correctness, receiver security and (statistical) sender security.

5 Semi-honest One-Query Co-PIR

We first present a construction of a one-query co-PIR that achieves semi-honest statistical sender security.

Before presenting our scheme we show how we can convert a LHE scheme over $\mathbb{Z}_2$ that supports ciphertext shrinking into an LHE scheme over $\mathbb{F}_q$ where $q = 2^\mu$ for some $\mu = \mathsf{poly}(\lambda)$. Here, we rely on the fact that multiplication over $\mathbb{F}_q$ can be expressed as a linear function over the field $\mathbb{Z}_2$. That is, suppose that an element $x \in \mathbb{F}_q$ is of the form $x = x_1 + x_2\alpha + \cdots + x_\mu\alpha^{\mu-1}$ where each $x_i \in \mathbb{Z}_2$ and α is a symbol. Then, for elements $a, x \in \mathbb{F}_q$ the product

$$xa = f_{1,a}(\mathbf{x}) + f_{2,a}(\mathbf{x})\alpha + \cdots + f_{\mu,a}(\mathbf{x})\alpha^{\mu-1}$$

where $\mathbf{x} = (x_1, \ldots, x_\mu)$ and each $f_{i,a} : \mathbb{Z}_2^\mu \to \mathbb{Z}_2$ is a $\mathbb{Z}_2$-linear function which depends solely on a. This means that there is a square matrix $\mathbf{A}$ (determined by a) such that the coefficients of the product $x \cdot a$ over $\mathbb{F}_q$ are the coefficients of $\mathbf{x} \cdot \mathbf{A}$ over $\mathbb{Z}_2$.

We now define the following functions.

- $\mathsf{FieldMult}(a \in \mathbb{F}_q, \mu)$ takes as input an element $a \in \mathbb{F}_q$ where $q = 2^\mu$. It outputs a matrix $\mathbf{A} \in \{0,1\}^{\mu \times \mu}$ such that the coefficients of $\mathbf{xA} \in \{0,1\}^\mu$ correspond to the coefficients of $x \cdot a$ over $\mathbb{F}_q$ (here $\mathbf{x}$ is a binary vector whose coefficients are the ones of $x \in \mathbb{F}_q$).

Ingredients. We need the following ingredients: Let $k, m \in \mathsf{poly}(\lambda)$, $\mu > \lceil \log m \rceil$ and $q = 2^\mu$. Let

- $\mathsf{LHE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Shrink}, \mathsf{DecShrink})$ be a rate-1 packed LHE scheme over $\mathbb{Z}_2$.
- $\mathsf{bin} : [m] \to \{0,1\}^\mu$ be the function that outputs the binary decomposition.

Construction 1. *We now present the full construction.*
$\mathsf{Query}(1^\lambda, i^* \in [m])$*:*

- *Compute* $\mathbf{i}^* = (i_1^*, \ldots, i_\mu^*) \leftarrow \mathsf{bin}(i^*)$.
- *For all* $\ell \in [\mu]$ *set*

$$\mathbf{T}_\ell^* = i_\ell^* \cdot \mathbf{I}_\ell = \begin{pmatrix} i_\ell^* & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & i_\ell^* & \ldots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & i_\ell^* \end{pmatrix} \in \{0,1\}^{k \times k}$$

where $\mathbf{I}_k$ *is the identity matrix of size* k.

- *Create* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LHE.KeyGen}(1^\lambda, k)$.
- *For all* $\ell \in [\mu]$ *compute* $\mathsf{ct}_\ell \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, \mathbf{T}^*_\ell)$.[15]
- *Output* $\mathsf{copir}_1 = (\mathsf{pk}, \{\mathsf{ct}_\ell\}_{\ell \in [\mu]})$ *and* $\mathsf{st} = (\mathsf{sk}, i^*)$.

$\mathsf{Send}\left(\mathsf{copir}_1, \mathbf{D} \in \left(\mathbb{F}_q^k\right)^m\right)$*:*

- *Parse* copir_1 *as* $(\mathsf{pk}, \mathsf{ct})$. *Additionally, parse* $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_m)$ *where each* $\mathbf{D}_i = (d_{i,1}, \ldots, d_{i,k}) \in \mathbb{F}_q^k$ *and* $d_{i,j} \in \mathbb{F}_q$ *for all* $j \in [k]$.
- *For all* $i \in [m]$ *and* $j \in [k]$ *determine* $\mathbf{A}_{i,j} \leftarrow \mathsf{FieldMult}(d_{i,j}, \mu)$. *Parse* $\mathbf{A}_{i,j} = \begin{pmatrix} - \mathbf{a}_{i,j,1} - \\ \vdots \\ - \mathbf{a}_{i,j,\mu} - \end{pmatrix} \in \{0,1\}^{\mu \times \mu}$.
- *For all* $i \in [m]$, $j \in [k]$ *and* $\ell \in [\mu]$ *compute* $\mathbf{C}_{i,j,\ell} \leftarrow \mathsf{SingleRowMatrix}(k, \mu, j, \mathbf{a}_{i,j,\ell})$.
- *For all* $i \in [m]$, *compute* $\mathbf{e}_i = (e_{i,1}, \ldots, e_{i,\mu}) \leftarrow \mathsf{bin}(i)$. *Additionally for all* $\ell \in [\mu]$ *set* $\mathbf{U}_{i,\ell} = \mathsf{Diag}(k, e_{i,\ell})$.
- *For all* $i \in [m]$ *consider the following* $\mathbb{Z}_2$ *function* $f_i : (\{0,1\}^{k \times k})^\mu \to \{0,1\}^{k \times \mu}$ *defined by*

$$f_i(\mathbf{X}_1, \ldots, \mathbf{X}_\mu) = \sum_{j=1}^{k} \sum_{\ell=1}^{\mu} (\mathbf{U}_{i,\ell} - \mathbf{X}_\ell) \cdot \mathbf{C}_{i,j,\ell}.$$

- *For all* $i \in [m]$ *compute* $\tilde{\mathsf{ct}}_i \leftarrow \mathsf{LHE.Eval\&Shrink}(\mathsf{pk}, f_i, (\mathsf{ct}_1, \ldots, \mathsf{ct}_\mu))$.
- *Output* $\mathsf{copir}_2 = \{\tilde{\mathsf{ct}}_i\}_{i \in [m]}$.

$\mathsf{Rec}(\mathsf{copir}_2, \mathsf{st})$*:*

- *Parse* copir_2 *as* $\{\tilde{\mathsf{ct}}_i\}_{i \in [m]}$ *and* st *as* (sk, i^*).
- *For all* $i \in [m] \setminus \{i^*\}$ *compute* $\mathbf{W}_i \leftarrow \mathsf{LHE.Dec}(\mathsf{sk}, \tilde{\mathsf{ct}}_i)$. *For each* $j \in [k]$ *parse each row* $\mathbf{w}_{i,j} \in \{0,1\}^\mu$ *of* $\mathbf{W}_i$ *as an element* $w_{i,j} \in \mathbb{F}_q$.
- *For all* $i \in [m] \setminus \{i^*\}$, *set* $\mathbf{e}_i \leftarrow \mathsf{bin}(i)$. *Parse* $\mathbf{e}_i$ *and* $\mathbf{i}^*$ *as* $\mathbb{F}_q$ *elements (that is,* $\hat{e}_i, \hat{i}^* \in \mathbb{F}_q$ *are the elements whose coefficients correspond to the coefficients of* $\mathbf{e}_i, \mathbf{i}^* \in \{0,1\}^\mu$*). Compute* $\tilde{\mathbf{D}}_i = (\hat{e}_i - \hat{i}^*)^{-1} \cdot (w_{i,1}, \ldots, w_{i,k})$ *over* $\mathbb{F}_q$. *Note that* $\tilde{\mathbf{D}}_i \in \mathbb{F}_q$.
- *Output* $\tilde{\mathbf{D}} = (\tilde{\mathbf{D}}_1, \ldots, \tilde{\mathbf{D}}_{i^*-1}, \mathbf{0}, \tilde{\mathbf{D}}_{i^*+1}, \ldots, \tilde{\mathbf{D}}_m)$.

The analysis of the scheme is presented in the full version of the paper.

6 Bit One-Query Co-PIR from Block One-Query Co-PIR

This construction is presented in the full version of the paper. The main idea behind it is that part of the erased block can be transmitted to the receiver via a PIR without incuring in additional communication. Some amount of care needs to be taken in order to preserve the PIR-compatibility of the scheme. We prove that the transformation preserves semi-honest security but it also preserves statistical sender security if the underlying PIR is statistical sender secure.

[15] Recall that an ecryption of a matrix is defined as individual packed encryptions of each column.

7 Semi-Honest Co-PIR from Semi-Honest One-Query Co-PIR

We now show how to bootstrap a one-query co-PIR into a multiple-query co-PIR. This construction works by recursing the one-query co-PIR multiple times and, at each step, one position of the database is erased. Since the underlying one-query co-PIR has rate 1, then there is no blowup in communication when we recurse it.

Construction 2. *Let* $\{\mathbb{H}_i\}_{i\in[t]}$ *be groups. For* $i \in [t]$ *let* $\mathsf{1QCoPIR}^{(i)} = (\mathsf{Query}, \mathsf{Send}, \mathsf{Rec})$ *be a one-query co-PIR scheme such that the outputs of* $\mathsf{1QCoPIR}^{(i)}.\mathsf{Send}$ *are of the form* $(\mathsf{head}, \alpha_1, \ldots, \alpha_m)$ *where* $\alpha_j \in \mathbb{H}_i$.

$\mathsf{Query}(1^\lambda, S)$*:*

- *Parse* $S = \{a_1, \ldots, a_t\}$.
- *For* $i \in [t]$ *compute* $(\mathsf{copir}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{1QCoPIR}^{(i)}.\mathsf{Query}(1^\lambda, a_i)$.
- *Output* $\mathsf{copir}_1 = \{\mathsf{copir}_{1,i}\}_{i\in[t]}$ *and* $\mathsf{st} = \{\mathsf{st}_i\}_{i\in[t]}$.

$\mathsf{Send}(\mathsf{copir}_1, \mathbf{D} \in \mathbb{H}^m)$*:*

- *Parse* copir_1 *as* $\{\mathsf{copir}_{1,i}\}_{i\in[t]}$. *Set* $\mathsf{DB}_0 = \mathbf{D}$.
- *For* $i \in [t]$ *do the following:*
  - *Compute* $\mathsf{copir}_{2,i} \leftarrow \mathsf{1QCoPIR}^{(i)}.\mathsf{Send}(\mathsf{copir}_{1,i}, \mathsf{DB}_{i-1})$.
  - *Parse* $\mathsf{copir}_{2,i}$ *as* $(\mathsf{head}_i, \alpha_{i,1}, \ldots, \alpha_{i,m})$.
  - *Set* $\mathsf{DB}_i = (\alpha_{i,1}, \ldots, \alpha_{i,m})$.
- *Set* $\mathsf{DB}^* = \mathsf{DB}_t$.
- *Output* $\mathsf{copir}_2 = (\mathsf{DB}^*, \mathsf{head}_1, \ldots, \mathsf{head}_t)$.

$\mathsf{Rec}(\mathsf{copir}_2, \mathsf{st})$*:*

- *Parse* copir_2 *as* DB^* *and* st *as* $\{\mathsf{st}_i\}_{i\in[t]}$.
- *Set* $\mathsf{DB}'_t = \mathsf{DB}^*$.
- *For* $i = t$ *to* 1, *set* $\mathsf{copir}'_{2,i} = (\mathsf{head}_i, \mathsf{DB}'_i)$ *and compute* $\mathsf{DB}'_{i-1} \leftarrow \mathsf{1QCoPIR}^{(i)}.\mathsf{Rec}(\mathsf{copir}'_{2,i}, \mathsf{st}_i)$.
- *Output* $\tilde{\mathbf{D}} = \mathsf{DB}'_0$.

The analysis of the scheme is presented in the full version of the paper.

8 Conditional Disclosure of Secrets for DDH-Based Encryption

In this section we present a black-box construction of a CDS for a specific language. Namely, our scheme guarantees that an El-Gamal public key is well-formed and that a certain ciphertext encrypts a bit. The scheme fulfills statistical sender privacy.

The main idea of this construction is to use AR codes to guarantee that the receiver's message is wellformed. The full construction and analysis is presented in the full version of the paper.

9 Statistical Sender Secure Co-PIR

In this section we present a scheme for statistical sender secure co-PIR. Our scheme works by bootstrapping a semi-honest co-PIR into a statistical sender secure one. We also show in the full version of the paper an alternative construction for SSP co-PIR from SSP PIR, albeit at the cost of slightly worse overall computational complexity.

We now show how to boostrap a semi-honest co-PIR into a statistical sender secure co-PIR using a CDS. Essentially, the CDS will ensure that the first message of the receiver is well-formed.

Let CoPIR be a semi-honest co-PIR scheme parametrized by m. Consider the following languange $\mathcal{L}_{\mathsf{CoPIR}}$ parametrized by CoPIR

$$\mathcal{L}_{\mathsf{CoPIR}} = \left\{ \mathsf{copir}_1 : \exists (S, r) \in [m]^t \times \{0,1\}^\lambda \text{ s.t. } \mathsf{copir}_1 \leftarrow \mathsf{CoPIR.Query}(1^\lambda, S; r) \right\}.$$

Clearly this is a NP languange thus there exists a statistical sender secure CDS scheme for this particular languange [33,44].

Ingredients. Let $\mathbb{H}$ be a group. Let

- $\mathsf{CoPIR} = (\mathsf{Query}, \mathsf{Send}, \mathsf{Rec})$ be a co-PIR scheme where the outputs of $\mathsf{CoPIR.Send}$ are of the form $(\alpha_1, \ldots, \alpha_m)$ where $\alpha_i \in \mathbb{H}$.
- $\mathsf{CDS} = (\mathsf{Enc}, \mathsf{Send}, \mathsf{Release})$ be a statistical sender secure CDS scheme for the the language $\mathcal{L}_{\mathsf{CoPIR}}$. Looking ahead, we will use the CDS construction from Sect. 8 to obtain a black-box construction.

Construction 3. *We now describe the construction in full detail.*
$\mathsf{Query}(1^\lambda, S)$*:*

- *Parse* $S = \{a_1, \ldots, a_t\}$.
- *Compute* $(\mathsf{copir}'_1, \mathsf{st}') \leftarrow \mathsf{CoPIR.Query}(1^\lambda, S; r)$ *using random coins* $r \in \{0,1\}^\lambda$.
- *Compute* $(\mathsf{cds}_1, \mathsf{st}'') \leftarrow \mathsf{CDS.Enc}(1^\lambda, (S, r))$.
- *Output* $\mathsf{copir}_1 = (\mathsf{copir}'_1, \mathsf{cds}_1)$ *and* $\mathsf{st} = (\mathsf{st}', \mathsf{st}'')$.

$\mathsf{Send}(\mathsf{copir}_1, \mathbf{D} \in \mathbb{G}^m)$*:*

- *Parse* copir_1 *as* $(\mathsf{copir}'_1, \mathsf{cds}_1)$.
- *Compute* $\mathsf{copir}'_2 \leftarrow \mathsf{CoPIR.Send}(\mathsf{copir}_1, \mathbf{D})$.
- *Parse* copir'_2 *as* $(\alpha_1, \ldots, \alpha_m)$ *and set* $\mathsf{DB} = (\alpha_1, \ldots, \alpha_m)$.
- *For all* $i \in [t]$ *sample* $\beta_i \leftarrow_\$ \mathbb{H}$.
- *Set* $\mathsf{DB}^* = \mathsf{DB} + (\beta_1, \ldots, \beta_t, 0 \ldots, 0)$.
- *Compute* $\mathsf{cds}_2 \leftarrow \mathsf{CDS.Send}(\mathsf{cds}_1, \mathcal{L}_{\mathsf{CoPIR}}, (\beta_1, \ldots, \beta_t))$.
- *Output* $\mathsf{copir}_2 = (\mathsf{DB}^*, \mathsf{cds}_2)$.

$\mathsf{Rec}(\mathsf{copir}_2, \mathsf{st})$*:*

- *Parse* copir_2 *as* $(\mathsf{DB}^*, \mathsf{cds}_2)$ *and* st *as* $(\mathsf{st}', \mathsf{st}'')$.
- *Compute* $(\beta'_1, \ldots, \beta'_t) \leftarrow \mathsf{CDS.Release}(\mathsf{cds}_2, \mathsf{st}'')$.

- *Compute* $\mathsf{DB}' \leftarrow \mathsf{DB}^* - (\beta'_1, \ldots, \beta'_t, 0, \ldots, 0)$.
- *Set* $\mathsf{copir}'_2 = \mathsf{DB}'$ *and compute* $\widetilde{\mathbf{D}} \leftarrow \mathsf{CoPIR.Rec}(\mathsf{copir}'_2, \mathsf{st}')$.
- *Output* $\widetilde{\mathbf{D}}$.

Due to space constrains, the analysis of the scheme is deferred to the full version of the paper.

10 Statistical Sender Private Oblivious Transfer with Optimal Rate

As an application for our statistical sender secure co-PIR scheme, we build an OT scheme. This OT scheme has overall rate 1 and achieves statistical sender privacy.

Before presenting our construction, we present some notation that we will use throughout this section.

- $\mathsf{RowMatrix}(\ell, n, \mathbf{v}_1, \ldots, \mathbf{v}_\ell)$: Takes row-vectors $\mathbf{v}_1, \ldots, \mathbf{v}_\ell \in \{0,1\}^n$ and outputs a matrix
$$\mathbf{V} = \begin{pmatrix} — \ \mathbf{v}_1 \ — \\ \vdots \\ — \ \mathbf{v}_\ell \ — \end{pmatrix},$$
i.e. for every $i \in [\ell]$ the i-th row of $\mathbf{V}$ is the row-vector $\mathbf{v}_i$.

Ingredients. We will need the following ingredients for our protocol:

- A PIR scheme $\mathsf{PIR} = (\mathsf{Query}, \mathsf{Send}, \mathsf{Retrieve})$.
- A (bit) co-PIR scheme $\mathsf{CoPIR} = (\mathsf{Query}, \mathsf{Send}, \mathsf{Rec})$ that is PIR-compatible parametrized by m.
- A rate-1 circuit-private LHE scheme $\mathsf{LHE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Shrink}, \mathsf{DecShrink})$ with plaintext space $\{0,1\}^\ell$ and for which shrinked ciphertexts have the form $\mathsf{ct} = (g, d_1, \ldots, d_\ell)$ where $g \in \mathbb{G}$ is a group element (for some large enough group $\mathbb{G}$, namely a DDH group) and $d_i \in \{0,1\}$.
- A download rate-1 CDS scheme $\mathsf{CDS} = (\mathsf{Enc}, \mathsf{Send}, \mathsf{Release})$ for the language
$$\mathcal{L} = \left\{ \mathsf{pk}, \{\mathsf{ct}_i\}_{i\in[\ell]} : \exists (r, \mathbf{s}_i, r_i) \text{ s.t. } \begin{array}{r} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LHE.KeyGen}(1^\lambda, \ell; r) \\ \mathbf{S}_i = \mathsf{SingleRowMatrix}(\ell, n, i, \mathbf{s}_i) \\ \mathsf{ct}_i \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, \mathbf{S}_i; r_i) \end{array} \right\}$$
for some $\mathbf{s}_i \in \{0,1\}^n$.
- The binary $\mathsf{LPN}(n, m, \rho)$ problem with dimension $n = \mathsf{poly}(\lambda)$, $m = n \cdot \ell \cdot \mathsf{poly}(\lambda)$ samples and slightly sub-constant noise-rate $\rho = m^{1-\epsilon}$.

Construction 4 (Optimal-rate SSP OT).

We now describe the scheme in full detail. $\mathsf{OTR}(1^\lambda, \mathbf{b} \in \{0,1\}^{m\ell})$:

- *Parse* $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_\ell)$*, where the* $\mathbf{b}_i \in \{0,1\}^m$ *are blocks of size* m*.*
- *Choose* $\mathbf{A} \leftarrow_\$ \{0,1\}^{n \times m}$ *uniformly at random and compute a pair of public and secret key* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{LHE.KeyGen}(1^\lambda, \ell; r)$ *using random coins* $r \in \{0,1\}^\lambda$*.*
- *For all* $i \in [\ell]$*, choose* $\mathbf{s}_i \leftarrow_\$ \{0,1\}^n$*, and* $\mathbf{e}_i \leftarrow_\$ \chi_{m,t}$*, compute* $\mathbf{c}_i \leftarrow \mathbf{s}_i\mathbf{A} + \mathbf{e}_i + \mathbf{b}_i$*, and set* $\mathbf{S}_i \leftarrow \mathsf{SingleRowMatrix}(\ell, n, i, \mathbf{s}_i)$*. Compute a matrix-ciphertext* $\mathsf{ct}_i \leftarrow \mathsf{LHE.Enc}(\mathsf{pk}, \mathbf{S}_i; r_i)$ *using random coins* $r_i \in \{0,1\}^\lambda$*.*
- *Compute* $(\mathsf{cds}_1, \tilde{\mathsf{st}}) \leftarrow \mathsf{CDS.Enc}(1^\lambda, w)$ *where* $w = (r, \{\mathbf{S}_i, r_i\}_{i\in[\ell]})$*.*
- *For all* $i \in [\ell]$ *set* $J_i = \mathsf{Supp}(\mathbf{e}_i)$ *to be the support of* $\mathbf{e}_i$*. Compute* $(\mathsf{copir}_{1,i}, \mathsf{st}_i) \leftarrow \mathsf{CoPIR.Query}(J_i)$*.*[16]
- *Output* $\mathsf{ot}_1 = \big(\mathsf{pk}, \mathbf{A}, \{\mathsf{ct}_i, \mathbf{c}_i, \mathsf{copir}_{1,i}\}_{i\in[\ell]}, \mathsf{cds}_1\big)$ *and* $\mathsf{st} = (\mathsf{sk}, \{\mathsf{st}_i, J_i\}_{i\in[\ell]}, \tilde{\mathsf{st}})$*.*

$\mathsf{OTS}(\mathsf{ot}_1, (\mathbf{m}_0, \mathbf{m}_1) \in (\{0,1\}^{m\ell})^2)$:

- *Parse* $\mathbf{m}_0 = (\mathbf{m}_{0,1}, \ldots, \mathbf{m}_{0,\ell})$ *and* $\mathbf{m}_1 = (\mathbf{m}_{1,1}, \ldots, \mathbf{m}_{1,\ell})$*, where each* $\mathbf{m}_{b,i} = (m_{b,i,1}, \ldots, m_{b,i,m}) \in \{0,1\}^m$*. Parse* $\mathsf{ot}_1 = \big(\mathsf{pk}, \mathbf{A}, \{\mathsf{ct}_i, \mathbf{c}_i, \mathsf{copir}_{1,i}\}_{i\in[\ell]}, \mathsf{cds}_1\big)$*.*
- *For* $i \in [\ell]$ *set* $\mathbf{z}_i = \mathbf{m}_{0,i}$*.*
- *Set* $\mathbf{Z} = \mathsf{RowMatrix}(\ell, m, \mathbf{z}_1, \ldots, \mathbf{z}_\ell)$*.*
- *For all* $i \in [\ell]$ *set* $\mathbf{C}_i = \mathsf{SingleRowMatrix}(\ell, m, i, \mathbf{c}_i)$ *and* $\mathbf{D}_i = \mathsf{Diag}(m, \mathbf{m}_{1,i} - \mathbf{m}_{0,i})$*.*
- *Define the* $\mathbb{Z}_2$*-linear function* $f : (\{0,1\}^{\ell\times n})^\ell \to \{0,1\}^{\ell\times m}$ *via*

$$f(\mathbf{X}_1, \ldots, \mathbf{X}_\ell) = \left(\sum_{i=1}^{\ell} (-\mathbf{X}_i\mathbf{A} + \mathbf{C}_i) \cdot \mathbf{D}_i\right) + \mathbf{Z}.$$

Additionally, define the $\mathbb{Z}_2$*-linear function* $g : (\{0,1\}^{\ell\times n})^\ell \to \{0,1\}^{\ell\times m}$ *via*

$$g(\mathbf{X}_1, \ldots, \mathbf{X}_\ell) = \left(\sum_{i=1}^{\ell} (-\mathbf{X}_i\mathbf{A} + \mathbf{C}_i + \mathbf{U}_i) \cdot \mathbf{D}_i\right) + \mathbf{Z}.$$

where $\mathbf{U}_i \leftarrow \mathsf{SingleRowMatrix}(\ell, m, i, \mathbf{1})$ *and* $\mathbf{1} = (1, \ldots, 1)$ *is the vector of length* m *which is* 1 *everywhere.*

- *Compute* $\mathsf{CT}_1 \leftarrow \mathsf{LHE.Eval\&Shrink}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)$ *and* $\mathsf{CT}_2 \leftarrow \mathsf{LHE.Eval\&Shrink}(\mathsf{pk}, g, \mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)$*.*
- *Parse* CT_1 *as* $\{g_i, d_{i,1}, \ldots, d_{i,\ell}\}_{i\in[m]}$ *where each* $g_i \in \mathbb{G}$ *and* $d_{i,j} \in \{0,1\}$*. Similarly parse* CT_2 *as* $\{h_i, f_{i,1}, \ldots, f_{i,\ell}\}_{i\in[m]}$ *each* $h_i \in \mathbb{G}$ *and* $f_{i,j} \in \{0,1\}$*.*
- *For all* $i \in [\ell]$ *set* $\mathbf{D}_i = (d_{1,i}, \ldots, d_{m,i})$ *and* $\mathbf{F}_i = (f_{1,i}, \ldots, f_{m,i})$*. Compute* $\mathsf{copir}_{2,i} \leftarrow \mathsf{CoPIR.Send}(\mathsf{copir}_{1,i}, \mathbf{D}_i)$ *and* $\mathsf{r}_i \leftarrow \mathsf{PIR.Send}(\mathsf{q}_i, \mathbf{F}_i)$ *where* $\mathsf{copir}_{1,i}$ *is parsed as the PIR message* q_i*.*
- *Set* $\mathsf{ot}_2' = \{g_i, \mathsf{copir}_{2,i}, h_i, \mathsf{r}_i\}_{i\in[\ell]}$*.*
- *Compute* $\mathsf{cds}_2 \leftarrow \mathsf{CDS.Send}(\mathsf{cds}_1, \mathcal{L}, \mathsf{ot}_2')$*.*
- *Output* $\mathsf{ot}_2 = \mathsf{cds}_2$*.*

[16] Recall that, since the CoPIR scheme is PIR-compatible then $\mathsf{copir}_{1,i}$ also corresponds to a first message PIR with input J_i.

$\mathsf{OTD}(\mathsf{ot}_2, \mathsf{st})$:

- *Parse* ot_2 *as* cds_2 *and* $\mathsf{st} = (\mathsf{sk}, \{\mathsf{st}_i, J_i\}_{i\in[\ell]}, \tilde{\mathsf{st}})$.
- *Compute* $\mathsf{ot}'_2 \leftarrow \mathsf{CDS.Release}(\mathsf{cds}_2, \tilde{st})$. *Parse* ot'_2 *as* $\{g_i, \mathsf{copir}_{2,i}, h_i, \mathsf{r}_i\}_{i\in[\ell]}$.
- *For all* $i \in [\ell]$ *compute* $\tilde{\mathbf{D}}_i = (\tilde{d}_{1,i}, \ldots, \tilde{d}_{m,i}) \leftarrow \mathsf{CoPIR.Retrieve}(\mathsf{copir}_{2,i}, \mathsf{st}_i)$.
- *Set* $\tilde{\mathsf{CT}}_1$ *to be* $\{g_i, \tilde{d}_{i,1}, \ldots, \tilde{d}_{i,\ell}\}_{i\in[m]}$. *Compute* $\mathbf{W} \leftarrow \mathsf{LHE.DecShrink}(\mathsf{sk}, \tilde{\mathsf{CT}}_1)$ *where* $\mathbf{W} \in \{0,1\}^{\ell\times m}$. *Parse* $\mathbf{W} = (w_{i,j})_{i\in[\ell],j\in[m]}$ *where* $w_{i,j} \in \{0,1\}$
- *For all* $i \in [\ell]$ *compute* $(v_{i,J_i[1]}, \ldots, v_{i,J_i[t]}) \leftarrow \mathsf{PIR.Retrieve}(\mathsf{r}_i, \mathsf{st}_i)$. *Additionally for all* $j \in [t]$, *compute* $y_{i,J_i[j]} \leftarrow \mathsf{LHE.Dec}(h_{J_i[j]}, v_{i,J_i[j]})$.
- *Set* $\mathbf{M} = (m_{i,j})_{i\in[\ell],j\in[m]} \in \{0,1\}^{\ell\times m}$ *where*

$$m_{i,j} = \begin{cases} y_{i,J_i[l]} \text{ if } l = J_i[l] \\ w_{i,j} \text{ otherwise} \end{cases} .$$

- *Output* $\mathbf{M}$.

The full analysis of the scheme is presented in the full version of the paper.

11 Two-Party Secure Computation with Overall Communication of $\mathcal{O}(|\mathcal{C}|) + \mathsf{poly}(\lambda)$

We now show an application of our optimal overall rate SSP OT. This application is in constructing a 2-Party secure computation (2PC) scheme with overall communication of $\mathcal{O}(|\mathcal{C}|) + \mathsf{poly}(\lambda)$, where $\mathcal{C}$ is the circuit to be computed and provides statistical semi-honest security against one of the parties and computational semi-honest security against the other party.

The protocol is just the classical GMW protocol [28] in the OT correlations model where the OT correlations are generated using our SSP OT.

Specifically, to compute a secret sharing of the AND of two wires whose values are themselves secret shared as (a_1, a_2) and (b_1, b_2) respectively, the parties first compute locally $a_1 \cdot b_1$ and $a_2 \cdot b_2$. One of the parties acts as the sender and the other acts as the receiver in two instances of 1-out-of-2 OT. Assume w.l.o.g. that P_1 acts as the sender and P_2 acts as the receiver. P_2 uses a_2 and b_2 as its choice bits and P_1 uses $(r_1, b_1 + r_1)$ and $(r_2, a_1 + r_2)$ respectively as the sender messages. P_2 obtains $a_2b_1 + r_1$ and $a_1b_2 + r_2$ as the outputs of the two OT executions. P_1 sets the share of the AND to be $a_1b_1 + r_1 + r_2$ and P_2 sets its share to be $a_2b_2 + a_1b_2 + r_1 + a_2b_1 + r_2$. Note that instantiating the GMW protocol with an OT scheme that does not have overall rate-1 incurs an communication complexity of $\mathsf{poly})|\mathcal{C}|, \lambda)$.

Lemma 13 ([28]). *Given a circuit* $\mathcal{C} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, *there exists a two-party* $O(|\mathcal{C}|)$*-round protocol in the OT-correlation model such that*

- *The protocol provides semi-honest statistical security.*
- *The communication complexity is upper-bounded by* $6|\mathcal{C}| + n + m + \mathsf{poly}(\lambda)$.
- *Both parties share* $2|\mathcal{C}|$ *OT correlations.*

For each gate, the parties need to perform 2 chosen input OTs. If they share 2 random OT correlations, these can be derandomized using the standard transformation from random OT to chosen input OT which takes 3 bits of communication per OT. Thus, the total communication is 6 bits per gate. If we setup the OT correlations using our SSP OT scheme, we obtain the following corollary.

Corollary 14. *Given a circuit* $\mathcal{C} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$*, there exists a two-party* α*-round protocol in the standard model such that*

- *The protocol is semi-honest secure against one of the parties and statistically semi-honest secure against the other one.*
- *The communication complexity is upperbounded by* $10|\mathcal{C}| + n + m + \mathsf{poly}(\lambda)$.

The $2 \cdot |\mathcal{C}|$ OT correlations can be shared using the OT scheme from Sect. 10 incurring in total communication approaching $4|\mathcal{C}| + \mathsf{poly}(\lambda)$ for large enough $|\mathcal{C}|$. Plugging this with the lemma above, we obtain a scheme with total communication $10|\mathcal{C}| + n + m + \mathsf{poly}(\lambda)$. Moreover, statistical security for one of the parties follow from the SSP property of the underlying OT.

Acknowledgements. Pedro Branco was partially funded by the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM research hub under grant number 16KISK038. Nico Döttling: Funded by the European Union (ERC, LACONIC, 101041207). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. Akshayaram Srinivasan was supported in part by a SERB startup grant and Google India Research Award.

References

1. Aggarwal, D., Döttling, N., Dujmovic, J., Hajiabadi, M., Malavolta, G., Obremski, M.: Algebraic restriction codes and their applications. In: Braverman, M. (ed.) 13th Innovations in Theoretical Computer Science Conference (ITCS 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 215, pp. 1–15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://drops.dagstuhl.de/opus/volltexte/2022/15598
2. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8
3. Applebaum, B.: Garbled circuits as randomized encodings of functions: a primer. In: Tutorials on the Foundations of Cryptography. ISC, pp. 1–44. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57048-8_1
4. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. In: 45th FOCS, pp. 166–175. IEEE Computer Society Press, October 2004
5. Badrinarayanan, S., Fernando, R., Jain, A., Khurana, D., Sahai, A.: Statistical ZAP arguments. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 642–667. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_22

6. Badrinarayanan, S., Garg, S., Ishai, Y., Sahai, A., Wadia, A.: Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part III. LNCS, vol. 10626, pp. 275–303. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_10
7. Badrinarayanan, S., Patranabis, S., Sarkar, P.: Statistical security in two-party computation revisited. In: Kiltz, E., Vaikuntanathan, V. (eds.) Theory of Cryptography. Lecture Notes in Computer Science, vol. 13748, pp. 181–210. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-22365-5_7
8. Bitansky, N., Freizeit, S.: Statistically sender-private OT from LPN and derandomization. Cryptology ePrint Archive, Paper 2022/185 (2022). https://eprint.iacr.org/2022/185
9. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018, pp. 896–912. ACM Press, October 2018
10. Boyle, E., et al.: Efficient two-round OT extension and silent non-interactive secure computation. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019, pp. 291–308. ACM Press, November 2019
11. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators: silent OT extension and more. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 489–518. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_16
12. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Correlated pseudorandom functions from variable-density LPN. In: 61st FOCS, pp. 1069–1080. IEEE Computer Society Press, November 2020
13. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators from ring-LPN. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12171, pp. 387–416. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56880-1_14
14. Boyle, E., Couteau, G., Meyer, P.: Sublinear secure computation from new assumptions. In: Kiltz, E., Vaikuntanathan, V. (eds.) Theory of Cryptography. Lecture Notes in Computer Science, vol. 13748, pp. 121–150. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-22365-5_5
15. Boyle, E., Gilboa, N., Ishai, Y.: Breaking the circuit size barrier for secure computation under DDH. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 509–539. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_19
16. Boyle, E., Gilboa, N., Ishai, Y.: Group-based secure computation: optimizing rounds, communication, and computation. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 163–193. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56614-6_6
17. Brakerski, Z., Branco, P., Döttling, N., Garg, S., Malavolta, G.: Constant ciphertext-rate non-committing encryption from standard assumptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 58–87. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64375-1_3
18. Brakerski, Z., Branco, P., Döttling, N., Pu, S.: Batch-OT with optimal rate. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology - EUROCRYPT 2022. Lecture Notes in Computer Science, vol. 13276, pp. 157–186. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-07085-3_6

19. Brakerski, Z., Döttling, N.: Two-message statistically sender-private OT from LWE. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 370–390. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03810-6_14
20. Brakerski, Z., Döttling, N., Garg, S., Malavolta, G.: Leveraging linear decryption: rate-1 fully-homomorphic encryption and time-lock puzzles. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 407–437. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36033-7_16
21. Brakerski, Z., Koppula, V., Mour, T.: NIZK from LPN and trapdoor hash via correlation intractability for Approximable relations. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 738–767. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56877-1_26
22. Chase, M., Garg, S., Hajiabadi, M., Li, J., Miao, P.: Amortizing rate-1 OT and applications to PIR and PSI. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part III. LNCS, vol. 13044, pp. 126–156. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90456-2_5
23. Döttling, N., Garg, S., Ishai, Y., Malavolta, G., Mour, T., Ostrovsky, R.: Trapdoor hash functions and their applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 3–32. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_1
24. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th FOCS, pp. 40–49. IEEE Computer Society Press, October 2013
25. Garg, S., Hajiabadi, M., Ostrovsky, R.: Efficient range-trapdoor functions and applications: rate-1 OT and more. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 88–116. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64375-1_4
26. Gentry, C., Halevi, S.: Compressible FHE with applications to PIR. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 438–464. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36033-7_17
27. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986). https://doi.org/10.1145/6490.6503
28. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC, pp. 218–229. ACM Press, May 1987
29. Goyal, V., Jain, A., Jin, Z., Malavolta, G.: Statistical zaps and new oblivious transfer protocols. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 668–699. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_23
30. Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Cryptol. **25**(1), 158–193 (2012)
31. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_9
32. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: a new representation with applications to round-efficient secure computation. In: 41st FOCS, pp. 294–304. IEEE Computer Society Press, November 2000
33. Ishai, Y., Kushilevitz, E.: Perfect Constant-round secure computation via perfect randomizing polynomials. In: Widmayer, P., Eidenbenz, S., Triguero, F., Morales, R., Conejo, R., Hennessy, M. (eds.) ICALP 2002. LNCS, vol. 2380, pp. 244–256. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45465-9_22

34. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography with constant computational overhead. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 433–442. ACM Press, May 2008
35. Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 575–594. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70936-7_31
36. Jain, A., Jin, Z.: Non-interactive zero knowledge from sub-exponential DDH. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 3–32. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_1
37. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 158–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_6
38. Kalai, Y.T., Khurana, D., Sahai, A.: Statistical witness indistinguishability (and more) in two messages. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 34–65. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_2
39. Kalai, Y.T., Lombardi, A., Vaikuntanathan, V., Wichs, D.: Boosting batch arguments and ram delegation. Cryptology ePrint Archive, Paper 2022/1320 (2022). https://eprint.iacr.org/2022/1320
40. Khurana, D., Mughees, M.H.: On statistical security in two-party computation. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part II. LNCS, vol. 12551, pp. 532–561. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64378-2_19
41. Khurana, D., Sahai, A.: How to achieve non-malleability in one or two rounds. In: Umans, C. (ed.) 58th FOCS, pp. 564–575. IEEE Computer Society Press, October 2017
42. Micciancio, D., Sorrell, J.: Simpler statistically sender private oblivious transfer from ideals of cyclotomic integers. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part II. LNCS, vol. 12492, pp. 381–407. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64834-3_13
43. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: Kosaraju, S.R. (ed.) 12th SODA, pp. 448–457. ACM-SIAM, January 2001
44. Ostrovsky, R., Paskin-Cherniavsky, A., Paskin-Cherniavsky, B.: Maliciously circuit-private FHE. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 536–553. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_30
45. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC, pp. 187–196. ACM Press, May 2008
46. Rabin, M.O.: How to exchange secrets with oblivious transfer. Cryptology ePrint Archive (2005)