# Simpler Statistically Sender Private Oblivious Transfer from Ideals of Cyclotomic Integers [⋆]

Daniele Micciancio and Jessica Sorrell

University of California San Diego, San Diego, USA
{daniele, jlsorrel}@cs.ucsd.edu

**Abstract.** We present a two-message oblivious transfer protocol achieving statistical sender privacy and computational receiver privacy based on the RLWE assumption for cyclotomic number fields. This work improves upon prior lattice-based statistically sender-private oblivious transfer protocols by reducing the total communication between parties by a factor $O(n \log q)$ for transfer of length $O(n)$ messages.
Prior work of Brakerski and Döttling uses transference theorems to show that either a lattice or its dual must have short vectors, the existence of which guarantees lossy encryption for encodings with respect to that lattice, and therefore statistical sender privacy. In the case of ideal lattices from embeddings of cyclotomic integers, the existence of one short vector implies the existence of many, and therefore encryption with respect to either a lattice or its dual is guaranteed to "lose" more information about the message than can be ensured in the case of general lattices. This additional structure of ideals of cyclotomic integers allows for efficiency improvements beyond those that are typical when moving from the generic to ideal lattice setting, resulting in smaller message sizes for sender and receiver, as well as a protocol that is simpler to describe and analyze.

## 1 Introduction

Oblivious transfer (OT) is a cryptographic primitive first introduced by Rabin [Rab05]. An OT protocol is carried out between two parties: a sender and a receiver. For our purposes, the sender possesses exactly two messages (binary strings), and the receiver possesses a bit corresponding to the sender's message that it wishes to receive. The protocol should satisfy security properties for both sender and receiver as well as a correctness property: the receiver should obtain the message corresponding to its bit with high probability while learning essentially nothing about the other message, and the sender should be unable to guess the receiver's bit with noticeable advantage.

[⋆] Research supported in part by the National Science Foundation (NSF) under grant CNS-1936703. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views, position or policy of the Government.

There are a variety of models for the parties involved in an OT protocol, as well as notions of security. In the semi-honest setting, the parties may be assumed to follow the protocol exactly, whereas in the malicious setting we require security even when either or both parties may deviate from the protocol. Using zero-knowledge proofs [GMW87], it is in fact possible to transform a semi-honest OT protocol into one secure against malicious parties, but given the overhead of this transformation, we will be interested in constructing malicious OT directly.

One of the strongest definitions of security we might hope to satisfy with our OT protocol is universally composable (UC) security. This definition requires that for any amount of deviation from the protocol, the outputs of both parties can still be efficiently simulated, even in an environment in which a variety of protocols are concurrently executed. This notion can already be achieved from standard lattice assumptions [PVW08], but requires a trusted third party to generate a common reference string during setup that may only be used a bounded number of times before this trusted setup must again be invoked. More recently, it has been shown how to compile an OT protocol satisfying a much weaker notion of security into one satisfying UC security [DGH$^+$20]. This weaker notion requires computational privacy for the receiver against a cheating sender, and only requires that a cheating receiver should not be able to output both of the sender's messages in their entirety. This compiler could potentially be used to give a UC-secure oblivious transfer protocol from lattice assumptions with a common reference string usable for an unbounded number of executions, but the compiled protocols are fairly complex and inefficient. In any case, it is known that a common reference string (and therefore a trusted third party) is required for any UC-secure OT protocol [CF01], and so other notions of security must be adopted in settings where no trusted party can be assumed.

Another notion, statistically sender-private OT (SSP OT), was introduced in [NP01][AIR01] and requires simulation security only against a cheating receiver, adopting a relaxed notion of computational security against a cheating sender. No setup is required to achieve this notion of security, and many constructions have been given from number theoretic assumptions [Kal05] [HK12] [BGI$^+$17]. In recent years, SSP OT constructions based on conjectured quantum-secure cryptographic assumptions have also begun to appear in the literature [BD18], [DGI$^+$19], and [BGI$^+$17] used with the results of [GH19] or [BDGM19].

Oblivious transfer has myriad uses in cryptography, and perhaps most notably is complete for secure multiparty computation (MPC) [Kil88][IPS09]. Since the security guarantees of the cryptographic constructions built from oblivious transfer depend very much on the properties of the underlying OT protocol, it is important to consider which cryptographic tasks motivate the study of SSP OT specifically. Badrinarayanan et al. [BGI$^+$17] used SSP OT to construct witness-indistinguishable arguments for NP, for which statistical sender privacy is required in the proof of zero knowledge. Jain et al. [JKKR17] showed that two-round SSP OT is sufficient to construct two-round delayed-input interactive arguments for NP that guarantee witness-indistinguishability, witness-hiding, and distributional weak zero-knowledge against delayed-input verifiers, though a computational notion of sender privacy also suffices for their constructions. Badrinarayanan et al. [BGJ$^+$18] subsequently showed that such arguments can be used to compile a three-round semi-malicious MPC protocol into a four-round malicious MPC protocol. SSP OT has been used in constructions of three-round concurrent MPC [BGJ$^+$17], two-message non-malleable commitments [KS17], and two-message witness-indistinguishable proof systems [KKS18]. It also has applications in fully-homomorphic encryption (FHE), as shown by Ostrovsky, Paskin-Cherniavsky, and Paskin-Cherniavsky [OPP14], in their construction of statistically circuit private FHE from SSP OT and any FHE scheme.

### 1.1 Related work

To compare prior work on statistically sender private two-round oblivious transfer, we will first need to introduce some convenient vocabulary for referring to the communication complexity of these protocols. We will be interested in the communication *rate* of existing protocols – the fractional bits of information transferred from sender to receiver per bit of communication in the protocol. Somewhat more quantitatively, the *overall rate* of a protocol that transfers a $\pi$ bit message to the receiver, requires $\nu$ bits of receiver communication and $\tau$ bits of sender communication is $\frac{\pi}{\nu+\tau}$. It is sometimes useful to distinguish between the contribution of the receiver's communication to the overall rate and the sender's. We refer to the former $(\pi/\nu)$ as the *upload rate*, and the latter $(\pi/\tau)$ as the *download rate*. Our protocol has (upload, download and overall) rate $O(1/\log(\lambda))$.

The first statistically sender private two-round oblivious transfer protocol based on lattice assumptions was given by [BD18], and we take this work as a starting point for our OT protocol. Brakerski and Döttling [BD18] gave a very nice generalization of an existing regularity lemma for lattices, and showed how to use this lemma along with duality properties of lattices to achieve statistically sender private OT with download rate $1/\log(\lambda)$ (similar to our protocol) but much worse upload (and overall) rate $1/(\lambda \cdot \mathsf{polylog}(\lambda))$.

There has since been significant progress in low-communication oblivious transfer from the Learning with Errors (LWE) assumption. Döttling et al. [DGI$^+$19] give a constant-rate SSP OT scheme via their construction of trapdoor hash functions. They use these functions to build download rate-$(1-O(1/\lambda))$ semi-honest OT, but with upload rate still inversely proportional to $\lambda$. They then observe that, for very long (polynomial in $\lambda$) messages, the upload rate (and therefore their overall rate) can be brought up to $1 - O(1/\lambda)$ by amortization. The sender's $\mathsf{poly}(\lambda)$-length strings may be divided up into blocks, and each block can be transferred using the receiver's first message. To achieve statistical sender privacy, they leverage a result from [BGI$^+$17], which gives a generic transformation from semi-honest OT with rate above $1/2$ to statistically sender private OT with constant rate. Though this improves on our protocol's $O(1/\log\lambda)$ overall rate, we observe that similar amortization may be applied to our protocol to achieve a constant upload rate by reuse of the receiver's first message, provided that the sender's messages are strings of length at least $\tilde{O}(n)$ (see Section 2.1, Lemma 1). However, many of the previously described applications of SSP OT do not call for polynomially large sender messages, and so the amortization method of [DGI$^+$19] is not applicable.

Furthermore, for applications requiring long sender messages in which statistical sender privacy can be relaxed to computational privacy, a different approach to amortization can be applied to both protocols. The receiver's first message is unchanged, but the sender will use the receiver's message to instead send one of two keys for a constant-rate symmetric encryption scheme, e.g., using a pseudorandom generator or a stream cipher. The sender may encrypt each of its two $\mathsf{poly}(\lambda)$-length messages under their respective keys, and send these ciphertexts as well. The receiver can then recover the key corresponding to its choice bit, and use this key to decrypt the longer message. The other key will be statistically hidden, by the SSP property of the OT protocol, however the sender's security will be reduced to that of the symmetric encryption scheme.

Badrinarayanan et al. [BGI$^+$17] also give a construction of constant rate SSP OT from any linear homomorphic encryption system with rate greater than 1/2. Such a homomorphic encryption system was later given by Gentry and Halevi [GH19], building off the GSW [GSW13] cryptosystem. Applied to the construction of [BGI$^+$17], their compressible FHE scheme allows compression of both the sender's and receiver's communication, but to achieve constant rate, the receiver must need super-linearly (in the security parameter) many $O(\lambda)$-length messages from the receiver. Concurrent work by Brakerski et al. [BDGM19] also gives a rate-1 FHE scheme based on a batched version [PVW08][BGH13] of the Regev [Reg05] encryption scheme. [BDGM19] also achieves high rate FHE via compression of multiple ciphertexts into a single ciphertext, and so will similarly require settings in which the sender's messages are of length polynomial in the security parameter to realize the benefit of compression.

### 1.2 Our Contribution

We give a simple, module lattice-based oblivious transfer protocol which improves over the overall rate of similar protocols ([BD18]) beyond the typical savings achieved when restricting to module lattices, saving a factor $O(\lambda \log \lambda)$. We compare our protocol to other lattice-based SSP OT protocols in two natural settings: a single execution of the protocol and $\lambda$ parallel executions. We show that for applications requiring at most $O(\lambda)$ messages of length $O(\lambda)$, we achieve the best known overall rate for SSP OT, giving significant improvements in both asymptotic and concrete parameters. Our protocol is also comparatively simple and efficient, requiring only a constant number of polynomial multiplications (see Section 4 for a more thorough comparison of the communication and computational complexity of this work with those of other SPP OT protocols).

### 1.3 Techniques

A lossy encryption scheme [KN08][PVW08] is a public-key encryption scheme that admits the generation of "lossy" public keys – public keys under which encryption statistically hides the encrypted message. The works of Peikert et al. [PVW08] and Hemenway et al. [HLOV11] show that, in fact, lossy encryption is equivalent to statistically sender private 2-round oblivious transfer.

Brakerski and Döttling [BD18] demonstrate one method for achieving such lossy encryption for lattice-based schemes, by showing that a single basis for a lattice can serve as both a lossy and lossless public key. In their OT protocol, the receiver sends a matrix $\mathbf{A}$, defining a q-ary lattice $\Lambda_q(\mathbf{A})$. The sender then encodes a string with respect to $\Lambda_q(\mathbf{A})$, and a second string with respect to the dual lattice $\Lambda_q^*(\mathbf{A})$. They show that for any valid $\mathbf{A}$, for a definition of validity that is efficiently verifiable, encoding with respect to at least one of $\Lambda_q(\mathbf{A})$ or $\Lambda_q^*(\mathbf{A})$ will statistically hide a constant fraction of the encoded string. The partially hidden string, given its encoding, will have sufficient min-entropy for the application of a randomness extractor, yielding a uniformly random one-time pad that can mask one of the sender's messages.

To provide some intuition as to why encoding with respect to both primal and dual lattices guarantees one of the two encodings will be somewhat lossy, we now describe their encoding method informally and at a high level. Encoding a string $m$ with respect to a lattice $\Lambda$ consists of injectively mapping $m$ to a lattice point $\mathbf{x} \in \Lambda$, and perturbing this lattice point by discrete Gaussian error $\mathbf{e}$ to produce a new point $\mathbf{t} = \mathbf{x} + \mathbf{e}$. Because $\mathbf{e}$ is drawn from a discrete Gaussian, a maximum likelihood decoding of $\mathbf{t}$ will identify the vector $\mathbf{x}' \in \Lambda$ that minimizes $\|\mathbf{e}\|_2 = \|\mathbf{t} - \mathbf{x}'\|_2$, and given a short basis for $\Lambda^*$, this can be done efficiently.

To see why such an encoding could be lossy for some lattices, consider the result of maximum likelihood decoding when $\|\mathbf{e}\|_2$ is much larger than the minimum distance $\lambda_1$ of the lattice, $\lambda_1(\Lambda) = \min_{\mathbf{v} \in \Lambda} \|\mathbf{v}\|_2$. In this case, there will be several candidate lattice points $\mathbf{x}'$ that correspond to similarly probable values for $\mathbf{e}$. This means that the most probable $\mathbf{x}'$ is not overwhelmingly likely to be correct – there is some entropy in at least one dimension of $\mathbf{x}$, given $\mathbf{t}$.

That one of $\Lambda$ and $\Lambda^*$ must contain enough short vectors to guarantee sufficient min-entropy for extraction follows in principle from a transference theorem of Banaszczyk [Ban93]. This theorem implies that for a lattice $\Lambda$ of rank $n$, there must be at least $n$ linearly independent vectors in $\Lambda \cup \Lambda^*$ of euclidean length no more than $\sqrt{n}$. So if $\Lambda$ has no vectors of length less than $\sqrt{n}$, $\Lambda^*$ must have a basis $\mathbf{B}$ for which $\max_{\mathbf{v} \in \mathbf{B}} \|\mathbf{v}\|_2 \leq \sqrt{n}$. In this case, for a large enough Gaussian, encoding with respect to $\Lambda^*$ will be highly lossy. Less conditional min-entropy can be guaranteed in the more balanced case, however, when there may be a few short vectors in both $\Lambda$ and $\Lambda^*$. Statistical privacy for the sender is therefore limited by this intermediate case.

We show that applying these same principles restricted to ideal lattices for ideals of cyclotomic integers guarantees more lossiness in the worst case for the sender. The structure of these ideal lattices ensures that $\Lambda$ must either have many short vectors or none at all, limiting the extent to which $\Lambda$ and $\Lambda^*$ can be

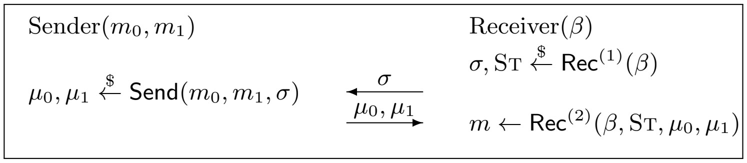
Sender($m_0, m_1$)
Receiver($\beta$)
$\sigma, \textsc{St} \xleftarrow{\$} \mathsf{Rec}^{(1)}(\beta)$
$\mu_0, \mu_1 \xleftarrow{\$} \mathsf{Send}(m_0, m_1, \sigma)$
$\xleftarrow{\sigma}$
$\xrightarrow{\mu_0, \mu_1}$
$m \leftarrow \mathsf{Rec}^{(2)}(\beta, \textsc{St}, \mu_0, \mu_1)$

Fig. 1: Two-message oblivious transfer protocol.

adversarially balanced by a cheating receiver. We exploit this structure to give a simpler statistically sender private OT protocol with smaller message sizes, yielding improvements in efficiency beyond those that are expected when moving from a generic lattice to ideal lattice scheme. The receiver's message, which dominates the communication complexity of [BD18], is reduced from $\mathcal{O}(n^2 \log^2 n)$ to $\mathcal{O}(n \log n)$ bits (see Figure 5), giving a $\mathcal{O}(\log n)$ factor improvement on top of the $\mathcal{O}(n)$ improvements typical of ideal lattice schemes. This is asymptotically modest, but as shown in Figure 6, yields significantly improved concrete parameters for lattice-based statistically sender secure oblivious transfer, even compared to other subsequent works.

## 2 Preliminaries

### 2.1 Oblivious Transfer

A *two-message oblivious transfer protocol*, $\mathsf{OT}$, comprises three algorithms:

$$\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$$

which are executed by two parties: a sender and a receiver. The protocol proceeds in stages as shown in Figure 1. (All algorithms additionally take a security parameter $1^\lambda$ as input, but we suppress this for notational convenience.) At the outset, the sender is given inputs $m_0, m_1 \in \{0,1\}^n$ for some fixed $n = \mathsf{poly}(\lambda)$, and the receiver is given input bit $\beta \in \{0,1\}$. The receiver runs $\mathsf{Rec}^{(1)}$ on its input $\beta \in \{0,1\}$. $\mathsf{Rec}^{(1)}$ then outputs a message $\sigma$ to the sender, and some state information $\textsc{St}$ to be passed to $\mathsf{Rec}^{(2)}$. On receiving $\sigma$, the sender runs $\mathsf{Send}(m_0, m_1, \sigma)$, which outputs a message pair $(\mu_0, \mu_1)$ to be transmitted to the receiver. In the final step, the receiver runs $\mathsf{Rec}^{(2)}(\beta, \textsc{St}, \mu)$ which returns a message in $\{0,1\}^n \cup \{\perp\}$.

We will be exclusively interested in two-message oblivious transfer protocols satisfying the following security and correctness properties.

**Definition 1 (Correctness).** *An* $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ *protocol is* correct *if for any pair of messages* $m_0, m_1$ *and bit* $b \in \{0,1\}$*,*

$$\Pr[\mathsf{Rec}^{(2)}(\mathsf{Send}(m_0, m_1, \mathsf{Rec}^{(1)}(b))) = m_b] \geq 1 - \epsilon$$

*for some negligible function* $\epsilon(n) = n^{-\omega(1)}$.

**Definition 2 (Statistical sender privacy).** *An* $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ *protocol is* statistically sender private *if there exists a potentially computationally unbounded extractor* $\mathtt{Ext}$ *such that for any receiver message* $\sigma$*,* $\mathtt{Ext}(\sigma)$ *outputs a bit* $b \in \{0, 1\}$ *such that for any pair of messages* $(m_0, m_1)$ *the two distributions*

$$\{\mathsf{Send}(\sigma, m_0, m_1)\} \approx_\Delta \{\mathsf{Send}(\sigma, m_b, m_b)\}$$

*are statistically close.*

Computational sender privacy is defined similarly, replacing statistical closeness $\approx_\Delta$ with computational indistinguishability. The main difference between sender privacy and full simulation security is that sender privacy does not require the bit $b$ to be efficiencly computable from $\sigma$. So, sender privacy can be described as a form of security with respect to a computationally unbounded simulator. For this reason, *statistical* security is perhaps a more natural requirement for the sender, and we do not consider computational sender privacy, except when discussing length extension techniques below.

**Definition 3 (Computational receiver privacy).** *An* $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ *protocol is* computationally receiver private *if the distributions* $\mathsf{Rec}^{(1)}(0)$ *and* $\mathsf{Rec}^{(1)}(1)$ *are computationally indistinguishable, i.e., for any (potentially cheating, probabilistic polynomial time) sender* $S^*$

$$|\Pr[S^*(\sigma) = 1 \mid \sigma \leftarrow \mathsf{Rec}^{(1)}(1)] - \Pr[S^*(\sigma) = 1 \mid \sigma \leftarrow \mathsf{Rec}^{(1)}(0)]| < \epsilon$$

*for some negligible function* $\epsilon(n) = n^{-\omega(1)}$*.*

Notice also that the sender security with efficient simulator (i.e., the ability to efficiently extract the bit $b$ from the receiver message $\sigma$) is clearly at odds with receiver security. In fact, two round protocols cannot achieve full simulation security, and goind beboynd sender privacy requires adding more communication rounds to the protocol.

As discussed in the introduction, it is possible to generically boost the upload rate of a statistically sender private oblivious transfer protocol from $1/\mathsf{poly}(\lambda)$ to a constant. Let $n(\lambda) \in \mathsf{poly}(\lambda)$, and let $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ be a statistically sender private oblivious transfer protocol with sender messages $m_0, m_1 \in \{0, 1\}^n$. Let $\ell(n) \in \mathsf{poly}(n)$. The protocol $\mathsf{OT}_\ell = (\mathsf{Rec}_\ell^{(1)}, \mathsf{Send}_\ell, \mathsf{Rec}_\ell^{(2)})$, described in Figure 2, transfers length $\ell(n)$ strings by reusing the output of $\mathsf{Rec}^{(1)}$ to execute $\ell/n$ parallel repetitions of the $\mathsf{Send}$ and $\mathsf{Rec}^{(2)}$ subroutines.

**Lemma 1 (Parallel OT execution).** *Let* $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ *be a statistically sender private oblivious transfer protocol with sender messages* $m_0, m_1 \in \{0, 1\}^n$*, upload rate* $\upsilon$*, and download rate* $\delta$*. Then for* $\ell(n) \in \mathsf{poly}(n)$*, the protocol* $\mathsf{OT}_\ell = (\mathsf{Rec}_\ell^{(1)}, \mathsf{Send}_\ell, \mathsf{Rec}_\ell^{(2)})$ *of Figure 2 is a statistically sender private oblivious transfer protocol with sender messages* $m_0, m_1 \in \{0, 1\}^{\ell(n)}$*, upload rate* $\upsilon\ell/n$*, and download rate* $\delta$*.*

**Algorithm 1** $\mathsf{Rec}_\ell^{(1)}$ Input: $\beta \in \{0,1\}$

```
$\sigma, \textsc{St} \leftarrow \mathsf{Rec}^{(1)}(\beta)$
return $\sigma$
```

**Algorithm 2** $\mathsf{Send}_\ell$ Input: $m_0, m_1 \in \{0,1\}^\ell$, $\sigma$

```
$m_0^{(1)} \| m_0^{(2)} \| \ldots \| m_0^{(\ell/n)} \leftarrow m_0$    Divide $m_0$ into blocks of length $n$
$m_1^{(1)} \| m_1^{(2)} \| \ldots \| m_1^{(\ell/n)} \leftarrow m_0$    Divide $m_1$ into blocks of length $n$
for $i \in \{1, \ldots, \ell/k\}$ do
    $\mu_0^{(i)}, \mu_1^{(i)} \leftarrow \mathsf{Send}(\sigma, m_0^{(i)}, m_1^{(i)})$
return $\{\mu_0^{(i)}, \mu_1^{(i)}\}_{i=1}^{\ell/n}$
```

**Algorithm 3** $\mathsf{Rec}_\ell^{(2)}$ Input: $\beta$, $\textsc{St}$, $(\mu_0, \mu_1)$

```
for $i \in \{1, \ldots, \ell/n\}$ do
    $m^{(i)} \leftarrow \mathsf{Rec}^{(2)}(\beta, \textsc{St}, \mu_0^{(i)}, \mu_1^{(i)})$
return $m^{(1)} \| m^{(2)} \| \ldots \| m^{(\ell/n)}$    Concatenate the $m^{(i)}$s
```

Fig. 2: Amortization of upload rate for an OT protocol for transfer of a single, $\mathsf{poly}(\lambda)$-length message.

*Proof.* The output of $\mathsf{Send}_\ell$ is by definition the same length as that of $\mathsf{Send}$, while the sender's messages are of length $\ell(n)$, and so the upload rate is $\upsilon\ell/n$. Both the output of $\mathsf{Send}_\ell$ and the length of the sender's messages have increased by a factor $\ell/n$ compared to $\mathsf{Send}$, and so the upload rate remains the same. Statistical sender privacy is preserved for a setting of $\ell(n) \in \mathsf{poly}(n)$, by a hybrid argument on the distributions of $(\mu_0^{(1)}, \mu_0^{(2)}, \ldots, \mu_0^{(\ell/n)})$ and $(\mu_1^{(1)}, \mu_1^{(2)}, \ldots, \mu_1^{(\ell/n)})$.

It is also possible to generically boost a statistically sender private OT protocol to one with constant overall rate, by trading statistical sender privacy for computational. Given a statistically sender private OT protocol $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ and a pseudorandom generator $\mathtt{G}$ with sufficiently large stretch, the protocol $\mathsf{OT}_\mathtt{G} = (\mathsf{Rec}_\mathtt{G}^{(1)}, \mathsf{Send}_\mathtt{G}, \mathsf{Rec}_\mathtt{G}^{(2)})$ shown in Figure 3 will have constant overall rate.

**Lemma 2 (OT Length extension).** *Let* $\mathsf{OT} = (\mathsf{Rec}^{(1)}, \mathsf{Send}, \mathsf{Rec}^{(2)})$ *be a statistically sender private oblivious transfer protocol with sender messages* $m_0, m_1 \in \{0,1\}^n$*, upload rate* $\upsilon$*, and download rate* $\delta$*. Let* $\ell(n) \in \mathsf{poly}(n)$ *and* $\mathtt{G}$ *be a pseudorandom generator with stretch* $\ell(n)$*. Then protocol* $\mathsf{OT}_\mathtt{G} = (\mathsf{Rec}_\mathtt{G}^{(1)}, \mathsf{Send}_\mathtt{G}, \mathsf{Rec}_\mathtt{G}^{(2)})$ *of Figure 3 is an oblivious transfer protocol with computational privacy for both*

**Algorithm 4** $\mathsf{Rec}^{(1)}_{\mathtt{G}}$ Input: $\beta \in \{0,1\}$

```
$\sigma, \textsc{St} \leftarrow \mathsf{Rec}^{(1)}(\beta)$
return $\sigma$
```

**Algorithm 5** $\mathsf{Send}_{\mathtt{G}}$ Input: $m_0, m_1 \in \{0,1\}^\ell$, $\sigma$

```
$s_0 \| s_1 \leftarrow \{0,1\}^{2\ell}$
$\mu_0, \mu_1 \leftarrow \mathsf{Send}(s_0, s_1, \sigma)$
$\mathtt{mask}_0 \leftarrow \mathtt{G}(s_0)$
$\mathtt{mask}_1 \leftarrow \mathtt{G}(s_1)$
return $(\mu_0, \mu_1, m_0 \oplus \mathtt{mask}_0, m_1 \oplus \mathtt{mask}_1)$
```

**Algorithm 6** $\mathsf{Rec}^{(2)}_{\mathtt{G}}$ Input: $\beta$, $\textsc{St}$, $(\mu_0, \mu_1, m_0 \oplus \mathtt{mask}_0, m_1 \oplus \mathtt{mask}_1)$

```
$s \leftarrow \mathsf{Rec}^{(2)}(\mu_0, \mu_1)$
$\mathtt{mask} \leftarrow \mathtt{G}(s)$
return $\mathtt{mask} \oplus \mu_\beta$
```

Fig. 3: Length extension of an OT protocol for transfer of a single, $\mathsf{poly}(\lambda)$-length message

*sender and receiver, sender messages* $m_0, m_1 \in \{0,1\}^{\ell(n)}$*, upload rate* $v\ell/n$*, and download rate at least* $(1 - n/\delta\ell)$*.*

*Proof.* The upload rate can be shown to be $v\ell/n$ as in Lemma 1. The output of $\mathsf{Send}_{\mathtt{G}}$ has increased over that of $\mathsf{Send}$ by an additive factor of $\ell$, and therefore the download rate is $\frac{\ell}{\ell+n/\delta} \geq 1 - n/\delta\ell$. The seed $s_{1-\beta}$ is statistically hidden from the receiver, and so $m_{1-\beta}$ is computationally hidden, with security reducing to the security of the pseudorandom generator $\mathtt{G}$ that was used as its mask.

### 2.2 Entropy and Extractors

For random variables $X, Y$, the *conditional min-entropy* of $X$ conditioned on $Y$ is

$$\mathbf{H}_\infty(X \mid Y) := -\log \max_{x,y} \Pr[X = x \mid Y = y].$$

[ILL89] show that a weak conditional min-entropy source $X$, along with a uniformly random seed $s$, can be used to generate an output distribution $\epsilon$-close to the uniform distribution, even given the seed $s$ and the possibly correlated value $Y$.

**Definition 4 ($(k, m, \epsilon)$-strong extractor).** *A function* $\mathsf{E} : \{0,1\}^l \times \mathcal{X} \to \{0,1\}^m$ *is a* $(k, m, \epsilon)$*-strong extractor (with* seed length $l$*) if for all random variables* $X$ *over* $\mathcal{X}$ *and* $Y$ *over* $\mathcal{Y}$ *such that* $\mathbf{H}_\infty(X \mid Y) \geq k$*, and for* $S$ *uniform on* $\{0,1\}^l$*, the statistical distance*

$$\Delta((\mathsf{E}(S, X), S, Y), (U_m, S, Y)) \leq \epsilon,$$

*where* $U_m$ *is the uniform distribution over* $\{0,1\}^m$*.*

There are many constructions of such $(k, m, \epsilon)$-strong extractors, all with varying seed lengths, codomain sizes, and runtimes. To ensure that our protocol's runtime is not asymptotically dominated by the application of the randomness extractor, we make use of a particular extractor from modified Toeplitz matrices, given by [Hay11]. This choice is more carefully justified in Section 3.5.

**Theorem 1 ([Hay11]).** *For any* $n, k \leq n$*, and* $\epsilon > 0$*, the following family of modified Toeplitz matrices over* $\mathbb{F}_q$ *is a* $(k, m, \epsilon)$*-strong extractor, for* $m = k - 2\log(1/\epsilon)$*, seed length* $l = \log q(n-1)$*, and input space* $\mathcal{X} = \mathbb{F}_q^n$*, running in time* $\mathcal{O}(n \log n)$*.*

*The seed* $s$ *selects a matrix* $\mathbf{M}$ *from the (implicitly defined) family as follows. Sample* $n - 1$ *elements* $x_i \in \mathbb{F}_q$ *using* $s$*. Define the matrix* $\mathbf{X} \in \mathbb{F}_q^{m \times n-m}$ *by* $\mathbf{X}_{i,j} = x_{n-m-j+i}$*. Let* $\mathbf{I}_m$ *be the* $m$*-dimensional identity matrix. Then the matrix* $\mathbf{M}$ *is*

$$\mathbf{M} = [\mathbf{X} \mid \mathbf{I}].$$

### 2.3 Lattices and Gaussian Measures

We write $[\mathbf{x}, \mathbf{y}]$ to indicate horizontal concatenation of vectors (or matrices) $\mathbf{x}$ and $\mathbf{y}$, and $(\mathbf{x}, \mathbf{y})$ to indicate vertical concatenation.

We define a *lattice* as a discrete additive subgroup of the space $\mathbb{R}^n$. A full-rank lattice of dimension $n$ is generated as all $\mathbb{Z}$-linear combinations of a set of $n$ linearly independent basis vectors in $\mathbb{R}^n$. When a basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ is specified, we write the lattice generated by $\mathbf{B}$ as

$$\Lambda(\mathbf{B}) = \{\mathbf{B}^t \mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$$

The $i$th successive minimum of a lattice $\Lambda$, for $1 \leq i \leq n$, is defined as

$$\lambda_i(\Lambda) = \min\{\lambda \in \mathbb{R}_{\geq 0} : \mathrm{rank}(\lambda \mathcal{B} \cap \Lambda) = i\}$$

where $\lambda \mathcal{B}$ denotes the ball of radius $\lambda$ centered on the origin. The *dual lattice* of $\Lambda$ is the set of vectors in $\mathbb{R}^n$ with integer inner product with all vectors of $\Lambda$, and is denoted $\Lambda^*$.

$$\Lambda^* := \{\mathbf{x} \in \mathbb{R}^n \mid \forall \mathbf{y} \in \Lambda : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$$

The Gaussian function $\rho_s : \mathbb{R}^n \to (0, 1]$ is $\rho_s(\mathbf{x}) = \exp(-\pi(\|\mathbf{x}\|/s)^2)$. We denote the Gaussian sum on a set $X \subset \mathbb{R}^n$ as $\rho_s(X) = \sum_{\mathbf{x} \in X} \rho_s(\mathbf{x})$. The *smoothing parameter* of a lattice, denoted by $\eta_\epsilon(\Lambda)$, is the smallest $s \in \mathbb{R}$ such that

$\rho_{1/s}(\Lambda^*) \leq 1+\epsilon$. We write $D_{\Lambda,s}$ to indicate the *discrete Gaussian distribution* of parameter $s$ over the points of lattice $\Lambda$, so that $D_{\Lambda,s}(\mathbf{x}) = \rho_s(\mathbf{x})/\rho_s(\Lambda)$.

We call a random variable $X$ or its distribution *subgaussian* over $\mathbb{R}$ of parameter $s$ if its tails are dominated by a Gaussian of parameter $s$, so that

$$\Pr\{|X| \geq t\} \leq 2e^{-\pi t^2/s^2} \text{ for all } t \geq 0.$$

A subgaussian variable $X$ with parameter $s > 0$ satisfies

$$\mathbb{E}[e^{2\pi t X}] \leq e^{\pi s^2 t^2}, \text{ for all } t \in \mathbb{R}.$$

The distribution $D_{\Lambda,s}$ is subgaussian with parameter $s$ for any lattice $\Lambda$ and $s > 0$, $s \in \mathbb{R}$. A random vector $\mathbf{x}$ of dimension $n$ is subgaussian of parameter $s$ if for all unit vectors $\mathbf{u} \in \mathbb{R}^n$, its one-dimensional marginals $\langle \mathbf{u}, \mathbf{x} \rangle$ are also subgaussian of parameter $s$. This extends to random matrices, where $\mathbf{X}^{m\times n}$ is subgaussian of parameter $s$ if for all unit vectors $\mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n$, $\mathbf{u}^t\mathbf{X}\mathbf{v}$ is subgaussian of parameter $s$. It follows immediately from these definitions that the concatenation of independent subgaussian vectors, all with parameter $s$, interpreted as either a vector or matrix, is also subgaussian with parameter $s$.

We will need the following tail bound on the length of a vector sampled from $D_{\Lambda,s}$.

**Lemma 3 ([Ban93]).** *For any $n$-dimensional lattice $\Lambda$ and $s > 0$, a point sampled from $D_{\Lambda,s}$ has Euclidean norm at most $s\sqrt{n}$, except with probability at most $2^{-2n}$.*

We will also need the following bounds on the smoothing parameter of any lattice $\Lambda$.

**Lemma 4 ([MR04]).** *For any $n$-dimensional lattice $\Lambda$, the smoothing parameter $\eta_{2^{-2n}}(\Lambda) \leq \sqrt{n}/\lambda_1(\Lambda^*)$.*

**Lemma 5 ([MR04]).** *For any $n$-dimension lattice $\Lambda$, and $\epsilon > 0$,*

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+2/\epsilon))}{\pi}}.$$

It follows from the above that for $\epsilon = 2^{-n}$, $\eta_\epsilon(\mathbb{Z}) \leq \sqrt{n}$.

The Poisson summation formula allows us to relate the Gaussian measure over a lattice to that over its dual.

**Lemma 6 (Poisson summation formula).** *For any lattice $\Lambda \subset \mathbb{R}^n$ and any complex-valued function $f : \mathbb{R}^n \to \mathbb{C}$, $f(\Lambda) = \frac{1}{\det(\Lambda)}\hat{f}(\Lambda^*)$.*

For $f = \rho_s$, it immediately follows from the above and the observation that $\hat{\rho}_s = s^n\rho_{1/s}$, that $\rho_s(\Lambda) = \frac{s^n}{\det(\Lambda)}\rho_{1/s}(\Lambda^*)$.

The following lemma of [BD18] gives a lower bound on the Gaussian measure over a lattice in terms of its successive minima.

**Lemma 7.** *For any $n$-dimensional lattice $\Lambda$, $k \in \mathbb{Z}$, $k \leq n$,*

$$\rho_s(\Lambda) \geq (s/\lambda_k(\Lambda))^k.$$

**Cyclotomic Integers and Module Lattices** Our protocol makes use of the structure of ideal lattices over cyclotomic integers. Let $\zeta_{2n}$ be a primitive $2n$th root of unity, for $n$ a power of 2. We denote by $\varPhi_{2n}(X)$ the $2n$th cyclotomic polynomial

$$\varPhi_{2n}(X) = \prod_{i \in \mathbb{Z}_{2n}^*} (X - \omega_{2n}^i) = X^n + 1,$$

which is the minimal polynomial of $\zeta_{2n}$, i.e. the lowest degree monic polynomial with coefficients in $\mathbb{Q}$ having $\zeta_{2n}$ as a root.

Our protocol operates on elements of the ring $\mathcal{R} = \mathbb{Z}[X]/(\varPhi_{2n}(X))$, and we write $\mathcal{R}_q$ to indicate the quotient ring $\mathcal{R}/q\mathcal{R}$. We embed elements of $\mathcal{R}$ into $\mathbb{Z}^n$ via the *coefficient embedding*, denoted $\sigma$, which takes an element $a \in \mathcal{R}$ to its coefficient vector. This embedding induces a geometry on $\mathcal{R}$, so that for any norm $\|\cdot\|$ defined on $\mathbb{Z}^n$, and any $a \in \mathcal{R}$, we take $\|a\| = \|\sigma(a)\|$. An ideal $\mathcal{I} \subset \mathcal{R}$ embeds under $\sigma$ as a lattice in $\mathbb{Z}^n$. Such a lattice $\varLambda = \sigma(\mathcal{I})$ is called an *ideal lattice.*

We may also use $\sigma$ to embed $k$-dimensional vectors over $\mathcal{R}$ into $\mathbb{Z}^{nk}$ by applying $\sigma$ element-wise, so that for $\mathbf{y} \in \mathcal{R}^k$, $\sigma(\mathbf{y}) = (\sigma(y_1), \ldots, \sigma(y_l))$. Let $\mathbf{A} \in \mathcal{R}_q^{l \times k}$ be generators of an $\mathcal{R}$ module $\mathbf{M} \subset \mathcal{R}^k$. Then we may define the *module lattice* $\varLambda = \sigma(\mathbf{M})$, suppressing the embedding notation, as

$$\varLambda(\mathbf{A}) = \{\mathbf{y} \in \mathcal{R}^k : \mathbf{y} = \mathbf{A}^t\mathbf{x},\ \mathbf{x} \in \mathcal{R}^l\}$$

We will also want to define two q-ary lattices in terms of $\mathbf{A} \in \mathcal{R}_q^{l \times k}$:

$$\varLambda_q(\mathbf{A}) = \{\mathbf{y} \in \mathcal{R}^k : \mathbf{y} = \mathbf{A}^t\mathbf{x} \bmod q\mathcal{R},\ \mathbf{x} \in \mathcal{R}^l\}$$

$$\text{and } \varLambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathcal{R}^k : \mathbf{A}\mathbf{x} = 0 \bmod q\mathcal{R}\}.$$

Note that

$$\varLambda_q^{\perp}(\mathbf{A})^* = \mathcal{R}_q^k + \{\tfrac{1}{q}\mathbf{A}^t\mathbf{s} : \mathbf{s} \in \mathcal{R}_q^k\} = \tfrac{1}{q}\varLambda(\mathbf{A}).$$

We will rely heavily on the following lemma on the successive minima of module lattices over a ring of cyclotomic integers. This is a well-established result (see, for instance, [FP11]), but we re-prove it here for completeness.

**Lemma 8.** *Let $\varLambda$ be a module lattice over $\mathcal{R}$. Then $\lambda_1(\varLambda) = \lambda_2(\varLambda) = \cdots = \lambda_n(\varLambda)$.*

*Proof.* Let $\mathbf{y} \in \mathcal{R}^k$, $\sigma(\mathbf{y}) \in \varLambda$, such that $\|\mathbf{y}\|_2 = \lambda_1(\varLambda)$. Then taking $\mathbf{y}^{(i)} = X^i\mathbf{y}$ for all $0 \leq i < n$, the multiplicative structure of $\mathcal{R}$ gives $\|\mathbf{y}^{(i)}\|_2 = \|X^i\mathbf{y}\|_2 = \|\mathbf{y}\|_2 = \lambda_1$. Suppose the $\mathbf{y}^{(i)}$ are not linearly independent. Then there exist $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{Z}$ such that

$$\alpha_0\mathbf{y} + \alpha_1X\mathbf{y} + \cdots + \alpha_{n-1}X^{n-1}\mathbf{y} = (\alpha_0 + \alpha_1X + \cdots + \alpha_{n-1}X^{n-1})\mathbf{y} = \mathbf{0} \in \mathcal{R}^k.$$

However, $\mathcal{R}$ is a Dedekind domain, and so this cannot be the case. Therefore the $\mathbf{y}^{(i)}$ and $\mathbf{y}$ are linearly independent and all of length $\lambda_1$.

The following is a corollary of Lemmata 8 and 7, taking $k = n$ in Lemma 7.

**Corollary 1.** *For any $m$-dimensional module lattice $\Lambda$ over $\mathcal{R}$,*

$$\rho_s(\Lambda) \geq (s/\lambda_1(\Lambda))^n.$$

The following lemma follows from techniques of [BD18] and [CDLP14].

**Lemma 9.** *Let $\Lambda' \subseteq \Lambda \subseteq \mathbb{Z}^n$ be lattices, and let $S$ be a symmetric set such that $\forall \mathbf{u} \in \Lambda$, $\mathbf{u}$ can be written uniquely as a sum $\mathbf{u} = \mathbf{x} + \mathbf{s}$, where $\mathbf{x} \in \Lambda'$ and $\mathbf{s} \in S$. Let $\mathbf{t} \in \mathbb{Z}^n$ and let $\sigma \in \mathbb{R}$. Then*

$$\frac{\rho_\sigma(\Lambda' + \mathbf{t})}{\rho_\sigma(\Lambda + \mathbf{t})} \leq \frac{1}{\rho_\sigma(S)}.$$

*Proof.*

$$\begin{aligned}
\rho_\sigma(\Lambda + \mathbf{t}) &= \sum_{\mathbf{x} \in \Lambda'} \sum_{\mathbf{s} \in S} \rho_\sigma(\mathbf{x} + \mathbf{t} + \mathbf{s}) \\
&= \sum_{\mathbf{x} \in \Lambda'} \sum_{\mathbf{s} \in S} \frac{1}{2}(\rho_\sigma(\mathbf{x} + \mathbf{t} + \mathbf{s}) + \rho_\sigma(\mathbf{x} + \mathbf{t} - \mathbf{s})) \\
&= \sum_{\mathbf{x} \in \Lambda'} \sum_{\mathbf{s} \in S} \frac{1}{2}(e^{-\pi\|\mathbf{x}+\mathbf{t}+\mathbf{s}\|^2/\sigma^2} + e^{-\pi\|\mathbf{x}+\mathbf{t}-\mathbf{s}\|^2/\sigma^2}) \\
&= \sum_{\mathbf{x} \in \Lambda'} \sum_{\mathbf{s} \in S} e^{-\pi\|\mathbf{x}+\mathbf{t}\|^2/\sigma^2} e^{-\pi\|\mathbf{s}\|^2/\sigma^2} (e^{-2\pi\langle \mathbf{x}+\mathbf{t}, \mathbf{s}\rangle/\sigma^2} + e^{2\pi\langle \mathbf{x}+\mathbf{t}, \mathbf{s}\rangle/\sigma^2}) \\
&\geq \sum_{\mathbf{x} \in \Lambda'} \rho_\sigma(\mathbf{x} + \mathbf{t}) \sum_{\mathbf{s} \in S} \rho_\sigma(\mathbf{s}) \\
&= \rho_\sigma(\Lambda' + \mathbf{t}) \rho_\sigma(S).
\end{aligned}$$

All proofs of correctness and security for our protocol hold for general cyclotomic rings of integers, beyond just power of 2 cyclotomics, by considering the canonical embedding of ring elements rather than the coefficient embedding described above. For the sake of simplicity, however, we restrict the description and analysis of the protocol to rings of integers for power of 2 cyclotomics only. In this case, one embedding gives a scaled isometry of the other, so either resulting lattice will have the structural properties we will require for lossy encryption. Other cyclotomic rings will give lattices that are distorted under the two choices of embedding, and so if other concerns force the use of this protocol in an alternative cyclotomic ring, the canonical embedding can be used instead.

### 2.4 Ring-LWE

The computational receiver privacy of our oblivious transfer protocol will rely on the RingLWE assumption for cyclotomic integers. Informally, it assumes that any probabilistic polynomial time adversary should have only negligible advantage distinguishing the RingLWE distribution described below from the uniform distribution over matrices with equivalent parameters.

**Definition 5** (RingLWE). *Let $\mathcal{R}$ be the $m$th cyclotomic ring of dimension $n = \varphi(m)$. Let $q \in \mathbb{Z}_{>0}$ and $\chi$ be a sub-Gaussian distribution over $\mathcal{R}$ with parameter $\alpha q$. The $\mathsf{RingLWE}_{q,\alpha}$ problem is to distinguish between independent samples of the form $(\mathbf{a}, s\mathbf{a} + \mathbf{e})$ for $s \leftarrow \chi$ fixed across samples, $\mathbf{a} \leftarrow \mathcal{R}_q^k$, and $\mathbf{e} \leftarrow \chi^k$, and the same number of samples of the form $(\mathbf{r}_0, \mathbf{r}_1)$, where each sample is chosen uniformly at random from $\mathcal{R}_q^k \times \mathcal{R}_q^k$.*

**Theorem 2 ([PRS17]).** *Let $K = \mathbb{Q}(\zeta_{2n})$ for $n$ a power of 2, and let $\mathcal{R}$ be the ring of integers of $K$. Let $\alpha = \alpha(n) \in (0,1)$, and let $q = q(n) \geq 2$ be an integer such that $\alpha q \geq 2\sqrt{n}$. There is a polynomial-time quantum reduction from $K$-$\mathsf{SIVP}_\gamma$ to (average-case, decision) $\mathsf{RingLWE}_{q,\alpha}$ for any $\gamma \leq \max\{\omega(\sqrt{n \log n}/\alpha), \sqrt{2}n\}$.*

## 3 Oblivious Transfer Protocol

We now present our $\mathsf{OT}$ protocol. In the following, let $\mathcal{R}$ denote the ring of integers of the $2n$-th cyclotomic number field for some $n$ a power of 2. Take $q = \mathsf{poly}(n)$ to be prime, $q \equiv 1 \mod 2n$. Let $s, \sigma_0, \sigma_1$ be Gaussian parameters and $\mathsf{E}$ be a $(\frac{3n}{2}, n, \epsilon)$-strong extractor for $\epsilon = 2^{-n/4}$, with seed length $l = 2n \log q - 1$, which is guaranteed to exist by Theorem 1. Lastly, take the sender's messages $m_0, m_1 \in \{0,1\}^n$, with $m_0$ encoded as an element of $\mathcal{R}_2$ and $\alpha \in \mathbb{Z}$ a parameter to be specified.

The protocol is described in Figure 4 and works as follows. The sender, on input two messages $m_0, m_1$, waits for the transmission of a matrix $\mathbf{A} \in \mathcal{R}_q^{2\times 3}$ from the receiver. Upon receiving $\mathbf{A}$, it uses this matrix to encrypt the two messages (in two different ways), and sends the resulting ciphertexts to the sender. The receiver, depending on the bit $b$, chooses the matrix $\mathbf{A}$ in such a way that it can decrypt either the first or the second message. It sends the matrix $\mathbf{A}$ to the sender, and when the sender returns the two ciphertexts, it uses $\mathbf{A}$ to decrypt the ciphertext of its choice.

Informally (and made formal in Section 3.3), the sender's privacy is preserved because one of the two sender encodings is statistically hidden. Identifying $\mathbf{x} + \Lambda_q^\perp[\mathbf{A}, \mathbf{I}]$ with $[\mathbf{A}, \mathbf{I}]\mathbf{x} \mod q$ for any $\mathbf{x} \in \mathbb{R}_q^5$ gives a bijective correspondence. So if the lattice $\Lambda_q^\perp[\mathbf{A}, \mathbf{I}]$ has many vectors that are short compared to the parameter of the Gaussian from which $\mathbf{x}_0$ is sampled, then following the intuition from Section 1.3, computing $[\mathbf{A}, \mathbf{I}]\mathbf{x}_0$ is a lossy encoding of $\mathbf{x}_0$. If it has enough short vectors, it will in fact lose (almost) *all* information about $\mathbf{x}_0$, so that the result is uniformly distributed over $\mathcal{R}_q^2$, hiding $m_0$. On the other hand, if $\Lambda_q[\mathbf{A}, \mathbf{I}] = \Lambda_q^\perp([\mathbf{I}, -\mathbf{A}^t])$ has many short vectors, then the same argument says that $[\mathbf{I}, -\mathbf{A}^t](\mathbf{x}_1, \mathbf{x}_2)$ is a lossy encoding of $\mathbf{x}_1$ and $\mathbf{x}_2$. For our settings of parameters, not all information about these vectors is lost, however, and so we use a randomness extractor applied to $\mathbf{x}_2$ to get a random mask, hiding $m_1$.

**Algorithm 7** $\mathsf{Rec}^{(1)}$

Input: $b \in \{0,1\}$

```
if $b = 0$ then
    $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q^3$
    $z \xleftarrow{\$} D_{\mathcal{R},s}$
    $\mathbf{e} \xleftarrow{\$} D^3_{\mathcal{R},s}$
    $\mathbf{b} = z \cdot \mathbf{a} + \mathbf{e}$
    $\mathbf{A} \leftarrow [\mathbf{a}, \mathbf{b}]^t$
    return $(\mathbf{A}, z)$
else
    $\bar{\mathbf{a}} \xleftarrow{\$} \mathcal{R}_q^2$
    $\mathbf{r} \xleftarrow{\$} D^2_{\mathcal{R},s}$
    $\mathbf{R} \xleftarrow{\$} D^{2\times 2}_{\mathcal{R},s}$
    $\mathbf{A} \leftarrow [\bar{\mathbf{a}} \mid \frac{q-1}{\alpha}\mathbf{I} + \bar{\mathbf{a}}\mathbf{r}^t + \mathbf{R}]$
    return $(\mathbf{A}, \mathbf{r})$
```

**Algorithm 8** $\mathsf{Send}$

Input: $\mathbf{A} \in \mathcal{R}_q^{2\times 3}$, $m_0, m_1 \in \mathcal{R}_2$

```
$\mathbf{x}_0 \xleftarrow{\$} D^5_{\mathcal{R},\sigma_0}$
$\mu_0 \leftarrow 2[\mathbf{A}, \mathbf{I}]\mathbf{x}_0 + \begin{bmatrix} 0 \\ m_0 \end{bmatrix} \mod q$
$\mathbf{x}_1 \xleftarrow{\$} D^3_{\mathcal{R},\sigma_1}$
$\mathbf{x}_2 \xleftarrow{\$} D^2_{\mathcal{R},\sigma_1}$
$\mathbf{c} \leftarrow \alpha \cdot (\mathbf{x}_1 - \mathbf{A}^t\mathbf{x}_2) \bmod q$
$r \leftarrow \{0,1\}^l$
$\mu_1 \leftarrow (\mathbf{c}, r, \mathsf{E}(r, \mathbf{x}_2 \bmod q) \oplus m_1)$
return $(\mu_0, \mu_1)$
```

**Algorithm 9** $\mathsf{Rec}^{(2)}$ Input: $b \in \{0,1\}$, $\textsc{St}$, $(\mu_0, \mu_1)$

```
if $b = 0$ then
    $z \leftarrow \textsc{St}$
    $m \leftarrow ([-z, 1]\mu_0 \bmod q) \bmod 2$
else
    $(\mathbf{c}, r, \tau) \leftarrow \mu_1$
    $\mathbf{r} \leftarrow \textsc{St}$
    $\mathbf{y} \leftarrow -(([\mathbf{r}, -\mathbf{I}] \cdot \mathbf{c}) \bmod q) \bmod \alpha$
    $m \leftarrow \mathsf{E}(r, \mathbf{y}) \oplus \tau$
return $m$
```

Fig. 4: Oblivious Transfer Protocol. In $\mathsf{Rec}^{(1)}$, the receiver generates a matrix along with auxiliary information that allows decoding of one of the sender's two messages. In $\mathsf{Send}$, the sender encodes its first message to be decodable with high probability if $A \leftarrow \mathsf{Rec}^{(1)}(0)$, and the second message so as to be decodable with high probability if $A \leftarrow \mathsf{Rec}^{(1)}(1)$. In the last stage, $\mathsf{Rec}^{(2)}$, the receiver decodes whichever of the sender's messages corresponds to its bit.

### 3.1 Correctness

In this section, we show that the OT protocol above satisfies our definition of correctness. The proof follows a standard argument for correctness of $\mathsf{RingLWE}$ cryptosystems, using concentration bounds to show that with high probability, the noise introduced by encryption does not exceed the threshold required for decoding.

**Lemma 10.** *If $s = 2\sqrt{n}$, $\sigma_0 \leq q/8\omega(\sqrt{(4ns^2+1)\log n})$ and $\sigma_1 \leq \alpha/2\omega(\sqrt{\log n})$ for $\alpha$ a power of 2 so that $\alpha \mid q-1$ and $\alpha \leq \sqrt{q-1}/s$, the protocol is correct.*

*Proof.* Since the entries of $\mathbf{e}$ and $z$ are chosen with gaussian distribution of parameter $s$, with all but negligible probability, $\beta_0 = \|[\mathbf{e}^t, -z]\|_2 < s\sqrt{4n}$. Similarly, the rows of $[\mathbf{r}, \mathbf{R}^t]$ have norm bounded by $\beta_1 < s\sqrt{3n}$ except with negligible probability, and we assume that both inequalities hold in the following.

We first consider the case that $b = 0$. In this case $\mathsf{Rec}(0, (\mu_0, \mu_1))$ computes

$$[-z, 1]\mu_0 = 2[\mathbf{e}^t, -z, 1]\mathbf{x}_0 + m_0 \pmod q$$

which equals $m_0$ modulo 2, as long as $\|[\mathbf{e}^t, -z, 1]\mathbf{x}_0\|_\infty < (q-1)/4$. Since the entries of $\mathbf{x}_0$ are subgaussian of parameter $\sigma_0$, the entries of $[\mathbf{e}^t, -z, 1]\mathbf{x}_0$ have subgaussian distribution of parameter

$$\sigma_0\sqrt{\beta_0^2+1} < \sigma_0\sqrt{4ns^2+1}.$$

So with all but negligible probability,

$$\|[\mathbf{e}^t, -z, 1]\mathbf{x}_0\|_\infty < \sigma_0\omega(\sqrt{(4ns^2+1)\log n}) \leq (q-1)/4.$$

We now consider the case that $b = 1$. The receiver will successfully recover $m_1 = \tau \oplus \mathsf{E}(r, \mathbf{x}_2)$ if $\mathbf{y} = \mathbf{x}_2$. By definition, before reduction modulo $\alpha$, the vector $\mathbf{y} \pmod q$ equals

$$\begin{aligned}
-(([\mathbf{r}, -\mathbf{I}] \cdot \mathbf{c}) &= -\alpha([\mathbf{r}, -\mathbf{I}]\mathbf{x}_1 - [\mathbf{r}, -\mathbf{I}]\mathbf{A}^t\mathbf{x}_2) \\
&= -\alpha([\mathbf{r}, -\mathbf{I}]\mathbf{x}_1) - [(q-1)\mathbf{I} + \alpha\mathbf{R}^t]\mathbf{x}_2 \\
&= (((1-q)\mathbf{x}_2 - \alpha([\mathbf{r}, -\mathbf{I}]\mathbf{x}_1 + \mathbf{R}^t\mathbf{x}_2)) \\
&= (\mathbf{x}_2 - \alpha([\mathbf{r}, -\mathbf{I}, \mathbf{R}^t](\mathbf{x}_1, \mathbf{x}_2))) \pmod q.
\end{aligned}$$

So, $\mathbf{y} = ((\mathbf{x}_2 - \alpha\mathbf{v}) \bmod q) \bmod \alpha$ for some vector

$$\mathbf{v} = [\mathbf{r}, -\mathbf{I}, \mathbf{R}^t](\mathbf{x}_1, \mathbf{x}_2).$$

We will show that, with high probability, $\|\mathbf{v}\|_\infty < (q-1)/(2\alpha)$ and $\|\mathbf{x}_2\|_\infty \leq \alpha/2$. It follows that, since $\mathbf{v}$ is an integer vector, we also have $\|\mathbf{v}\|_\infty \leq (q-1)/(2\alpha) - 1$, and

$$\|\mathbf{x}_2 - \alpha\mathbf{v}\|_\infty \leq \|\mathbf{x}_2\|_\infty + \alpha\|\mathbf{v}\|_\infty \leq \frac{\alpha}{2} + \frac{q-1}{2} - \alpha < \frac{q}{2}.$$

So, the computation of $\mathbf{y}$ recovers $\mathbf{v}$ over the integers, and $\mathbf{y} = \mathbf{v} \bmod \alpha = \mathbf{x}_2 \bmod \alpha = \mathbf{x}_2$.

Both $\mathbf{x}_1$ and $\mathbf{x}_2$ are drawn from a discrete Gaussian of parameter $\sigma_1$ and so, by an argument analogous to that of the previous case, the entries of $\mathbf{v}$ have subgaussian distribution of parameter

$$\sigma_1\sqrt{\beta_1^2+1} < \sigma_1\sqrt{3ns^2+1}.$$

Then with all but negligible probability we can bound the $\ell_\infty$ norm of the result by

$$\begin{aligned}\|\mathbf{v}\|_\infty &< \sigma_1\sqrt{3ns^2+1}\cdot\omega(\sqrt{\log n})\\ &\leq \frac{\alpha\sqrt{3ns^2+1}}{2}\\ &\leq \frac{q-1}{2\alpha}.\end{aligned}$$

We can also bound the coefficients of $\mathbf{x}_2$ by $\sigma_1 \cdot \omega(\sqrt{\log n}) \leq \alpha/2$ so the output is correct except with negligible probability.

### 3.2 Computational Receiver Privacy

Here we show that the receiver enjoys computational privacy. This follows immediately from the pseudorandomness of $\mathsf{RingLWE}$.

**Lemma 11.** *Let $q = \mathsf{poly}(n)$ be prime, $q \equiv 1 \mod 2n$. Take $s > 2\sqrt{n}$. Then, the distributions $\mathsf{Rec}^1(0)$ and $\mathsf{Rec}^1(1)$ are computationally indistinguishable under standard $\mathsf{RingLWE}$ assumptions.*

*Proof.* We show that the distribution of matrix $\mathbf{A}$ computed by both $\mathsf{Rec}^1(0)$ and $\mathsf{Rec}^1(1)$ is pseudorandom. For $\mathsf{Rec}^1(0)$, $\mathbf{A}^t = [\mathbf{a}, z\mathbf{a}+\mathbf{e}]$ is just the $\mathsf{RingLWE}$ distribution with gaussian parameter $s \geq 2\sqrt{n}$. For $\mathsf{Rec}^1(1)$, $[\bar{\mathbf{a}}, \bar{\mathbf{a}}\mathbf{r}^t + \mathbf{R}]$ is also the $\mathsf{RingLWE}$ distribution with secret $\mathbf{r}$ and noise $\mathbf{R}$. So, it is indistinguishable from the uniform distribution under standard $\mathsf{RingLWE}$ assumptions. Adding $[\mathbf{0}, \frac{q-1}{\alpha}\mathbf{I}]$ maps the uniform distribution to itself. So, it preserves indistinguishability.

### 3.3 Statistical Sender Privacy

Finally, we show statistical privacy for the sender. Recall that statistical privacy requires that for all inputs $\mathbf{A}$, one of the sender's two messages must be statistically hidden. As previously described, we wish to consider two cases: one in which $\Lambda_q^\perp([\mathbf{A},\mathbf{I}])$ has many short vectors, and one in which $\Lambda_q([\mathbf{A},\mathbf{I}])$ does, formalized in such a way that these cases are exhaustive and give the necessary guarantees on lossiness. To that end, we actually analyze the following two cases: one in which the smoothing parameter of $\Lambda_q^\perp([\mathbf{A},\mathbf{I}])$ is small compared to $\sigma_0$ ($\Lambda_q^\perp([\mathbf{A},\mathbf{I}])$ has many short vectors), and the other in which the smoothing parameter is large ($\Lambda_q([\mathbf{A},\mathbf{I}])$ has short vectors). In the first case, the sender's first message $m_0$ must be statistically hidden, and in the second, $m_1$ must be.

**Theorem 3.** *Assume $\sigma_0\sigma_1 \geq 8q\sqrt{5n}\omega(\sqrt{\log n})$ and $\sigma_1 \leq q/\sqrt{n}$. Then there exists an unbounded extractor $\mathtt{Ext}$ taking as input an element of $\mathcal{R}^{2\times 3}$ and outputting a bit $b$, such that for all $\mathbf{A} \in \mathcal{R}^{2\times 3}$, letting $b \leftarrow \mathtt{Ext}(\mathbf{A})$, it holds for all $m_0, m_1 \in \mathcal{R}_2$,*

$$\mathsf{Send}(\mathbf{A}, m_0, m_1) \approx_\Delta \mathsf{Send}(\mathbf{A}, m_b, m_b).$$

*Proof.* We consider two propositions, at least one of which must be true of $\Lambda(\mathbf{A})$, and show that in each case, one of $m_0$ or $m_1$ must be statistically hidden. It follows that we can change either $m_0$ to $m_1$ or $m_1$ to $m_0$, without affecting the distribution by a noticeable amount.

First consider the case $\sigma_0 > \eta_\epsilon(\Lambda_q^\perp([\mathbf{A},\mathbf{I}])) \cdot \omega(\sqrt{\log n})$. If this is the case, $[\mathbf{A},\mathbf{I}]\mathbf{x}_0$ is statistically close to uniform. Since $q$ is odd, multiplying by 2 and adding $(0, m_0)$ is a bijection, and preserves the uniform distribution. So, $\mu_0$ is independent of $m_0$. Clearly, $\mu_1$ is also independent of $m_0$.

Next we consider the case $\eta_\epsilon(\Lambda_q^\perp([\mathbf{A},\mathbf{I}])) \geq \sigma_0/2\omega(\sqrt{\log n})$. We show that $\mathbf{x}_2 \pmod q$ must have high min-entropy $H_\infty(\mathbf{x}_2 \mid \mathbf{c}) \geq 3n/2$ even when conditioned on $\mathbf{c}$. So, the output of the seeded extractor $\mathsf{E}$ is (statistically close to) a uniformly random $n$-bit mask, and $m_1$ is statistically hidden. Notice that the conditional distribution of $(\mathbf{x}_1, \mathbf{x}_2)$ given $\mathbf{c}$ is precisely $D_{C,\sigma_1}$ where

$$C = (\mathbf{c}/\alpha, \mathbf{0}) + \Lambda_q^\perp([\mathbf{I}, -\mathbf{A}^t]) = (\mathbf{c}/\alpha, \mathbf{0}) + \Lambda_q([\mathbf{A},\mathbf{I}]).$$

Since $\eta_\epsilon(\Lambda_q^\perp([\mathbf{A},\mathbf{I}])) \geq \sigma_0/2\omega(\sqrt{\log n})$ by assumption, and $\Lambda_q^*([\mathbf{A},\mathbf{I}])) = \frac{1}{q}\Lambda_q^\perp([\mathbf{A},\mathbf{I}])$, we have

$$\lambda_1(\Lambda_q([\mathbf{A},\mathbf{I}])) \leq \frac{q\sqrt{5n}}{\eta_\epsilon(\Lambda_q^\perp([\mathbf{A},\mathbf{I}]))} \leq \frac{2q\sqrt{5n} \cdot \omega(\sqrt{\log n})}{\sigma_0}.$$

Therefore from Corollary 1 we have that

$$\rho_{\sigma_1}(\Lambda_q([\mathbf{A},\mathbf{I}])) \geq \left(\frac{\sigma_1}{\lambda_1}\right)^n \geq \left(\frac{\sigma_0\sigma_1}{2q\sqrt{5n} \cdot \omega(\sqrt{\log n})}\right)^n \geq 4^n.$$

For any $\mathbf{c}$ and $\mathbf{x}^*$, let $X = \{(\mathbf{x}_1, \mathbf{x}_2) \in C \mid \mathbf{x}_2 = \mathbf{x}^* \pmod q\}$, and notice that $X$ is a coset $\mathbf{t} + q\mathcal{R}^5$ for some $\mathbf{t} \in C$. Let $S$ be the set of coset representatives of $\Lambda_q([\mathbf{A},\mathbf{I}])/q\mathcal{R}^5$ obtained by a "centered" reduction (so that all representative have coefficients in the range $(-q/2, q/2)$, recalling that $q$ is odd). Note that $S$ is a symmetric set and that any point $\mathbf{u} \in \Lambda_q([\mathbf{A},\mathbf{I}])$ can be uniquely written as the sum $\mathbf{u} = \mathbf{x} + \mathbf{s}$, where $\mathbf{x} \in q\mathcal{R}^5$ and $\mathbf{s} \in S$. We may then use Lemma 9 to conclude that

$$\begin{aligned} &\Pr\{(\mathbf{x}_2 = \mathbf{x}^*) \bmod q \mid (\mathbf{x}_1, \mathbf{x}_2) \leftarrow D_{C,\sigma_1}\} \\ &\qquad = \frac{\rho_{\sigma_1}(X)}{\rho_{\sigma_1}(C)} \leq \frac{1}{\rho_{\sigma_1}(S)}. \end{aligned}$$

A vector $\mathbf{u}$ sampled from a discrete gaussian over $\Lambda_q([\mathbf{A},\mathbf{I}])$ of parameter $\sigma_1$ must have $\|\mathbf{u}\|_\infty < q/2$ with probability at least $1 - 2^{-5n}$, so we have that

$$\rho_{\sigma_1}(S) \geq (1 - 2^{-5n}) \cdot \rho_{\sigma_1}(\Lambda_q([\mathbf{A},\mathbf{I}])) \geq (1 - 2^{-5n}) \cdot 4^n > 2^{2n-1}.$$

Therefore $\frac{1}{\rho_{\sigma_1}(S)} \leq 2^{-2n+1}$, and so $H_\infty(\mathbf{x}_2 \mod q \mid \mathbf{c}) \geq 3n/2$.

Finally, we must argue that there exists an unbounded extractor `Ext` that, on input $\mathbf{A}$, correctly identifies which of the cases above holds with its output $b$. We first observe that approximating the value of the smoothing parameter $\eta_\epsilon(\Lambda_q^\perp([\mathbf{A},\mathbf{I}]))$ to within a factor $(1 + o(1))$ can be done in deterministic

$2^{O(n)}\mathsf{polylog}(1/\epsilon)$ time and $2^{O(n)}$ space, as shown by Chung et al. [CDLP14]. Then the extractor that on input $\mathbf{A}$, runs the algorithm of [CDLP14], outputs 0 if $\eta_\epsilon(\Lambda_q^\perp([\mathbf{A},\mathbf{I}])) < \sigma_0/2\omega(\sqrt{\log n})$, and 1 otherwise, will satisfy our definition of statistical sender privacy.

### 3.4 Parameters

It remains to fix values for parameters that satisfy the competing demands of security and correctness. These require that

$$\sigma_0 \leq q/8\omega(\sqrt{(4ns^2+1)\log n}),$$

$$\sigma_1 \leq \alpha/2\omega(\sqrt{\log n}),$$

$$\alpha \leq \sqrt{q-1}/s,$$

and

$$\sigma_0\sigma_1 \geq 8q\sqrt{5n}\cdot\omega(\sqrt{\log n})$$

Letting $\gamma(n) \in \omega(\sqrt{\log n})$, a possible setting of parameters is $q \in \Theta(n^4\gamma^6(n))$, $s = 2\sqrt{n}$, $\alpha \in \Theta(n^{1.5}\gamma^3(n))$, $\sigma_0 \in \Theta(n^3\gamma^5(n))$ and $\sigma_1 \in \Theta(n^{1.5}\gamma^2(n))$.

### 3.5 Choice of Extractor

A reader already familiar with existing regularity lemmas for lattices may wonder about the use of a generic randomness extractor for producing a uniformly random string from $\mathbf{x}_2$. In the given protocol, $\mathbf{x}_2 \in \mathcal{R}^2$ is sampled from a discrete Gaussian with parameter $\sigma_1$, but the generic extractor cannot exploit this additional information about its input. If we instead sampled a matrix $\bar{\mathbf{A}}$ uniformly at random from $\mathcal{R}_q^{k\times l}$, and took $\mathbf{A} = [\mathbf{I}_k, \bar{\mathbf{A}}]$, then with overwhelming probability the distribution induced by $\mathbf{Ax}$ is statistically close to uniform over $\mathcal{R}_q^k$, for $\mathbf{x}$ sampled from $D_{\mathcal{R}^{k+l},\sigma}$ with $\sigma > 2nq^{k/(l+k)+2/n(l+k)}$, by a theorem of Lyubashevsky, et al. [LPR13]. This approach is arguably more natural, as it consists solely of ring operations and makes use of the distribution from which $\mathbf{x}_2$ is drawn.

However $\mathbf{x}_2$ comprises two elements of $\mathcal{R}$, which forces $l = k = 1$. Correctness and receiver security for the protocol require that $\sigma_1 < \sqrt{q}/\sqrt{n}\omega(\sqrt{\log n})$, and so $\sigma_1$ is not large enough to guarantee negligible distance from uniformity over $\mathcal{R}_q$. We may instead consider taking $k = 1, l = 4$, sampling $\mathbf{A} = [\mathbf{1}, \bar{\mathbf{A}}] \in \mathcal{R}^5$, and using $\mathbf{A}(\mathbf{x}_1, \mathbf{x}_2)$ as the mask for plaintext message $m_1$, however the Toeplitz matrix construction applied directly to $\mathbf{x}_2$ proves to be comparably efficient, without imposing additional constraints on parameter choices. The Toeplitz matrix sampled by the extractor is an element of $\mathbb{F}_q^{n\times 2n}$, and because this Toeplitz matrix multiplication can be performed at least as efficiently as polynomial multiplication of two degree $2n$ polynomials, there is no clear reason to prefer the more “natural” approach to the use of a more generic extractor.

## 4 Comparison to Related Protocols

In this section, we provide comparisons of our protocol to existing lattice-based SSP OT protocols. Specifically, we present the asymptotic and concrete parameters required by [BD18], [DGI$^+$19], [GH19], and this work, as well as communication and computational complexity for all protocols. We remark that, when transferring sufficiently long messages, in the computational setting, both the computational and communication costs of any OT protocol can be reduced to linear in the message length using standard techniques. Namely, one can use the OT protocol on two random (fixed length) strings $\mathbf{x}_0, \mathbf{x}_1 \in \{0,1\}^n$, and then use these random strings to encrypt the actual messages using a pseudorandom generator or stream cipher. So, for a meaningful comparison, we fix the length of the messages to be transfered to the security parameter $n$. So, we give comparisons in two representative settings which naturally arise in applications of SSP OT: a single execution of the protocol, and $O(n)$ parallel executions, all with sender messages of length $n$.

We note that the first setting is particularly unfavorable for SSP OT constructions that make use of compressible FHE. These protocols look more attractive in applications that require $\mathsf{poly}(n)$ simultaneous transfers. When many parallel OTs are required by an application, the receiver can compress fully-homomorphic encryptions of multiple bits and send the resulting compressed ciphertext along with a public key to the sender. The sender can then decompress the ciphertexts, homomorphically select the message corresponding to each of the encrypted bits, compress the resulting encryptions of its messages, and send a single compressed ciphertext to the receiver. But in the setting of a single execution of an OT protocol with $O(n)$-length sender messages, these constructions cannot take full advantage of the compressibility of the FHE scheme. In these cases, it should be possible for the sender to use a (not necessarily homomorphic) encryption scheme with more compact ciphertexts, by using key switching techniques. But even this will not improve the upload rate, however, which is the dominant contribution to the overall rate for these protocols.

The second comparison of these protocols is in a context closer to that of their applications [BGJ$^+$18] [JKKR17] [BGI$^+$17]. In these applications, linearly many parallel OT executions are required, and so the FHE-based OT schemes can actually make use of their compressibility. The $O(n^2)$ bits to be transferred in this case still fall short of allowing the amortization necessary for these protocols to achieve constant overall rate.

When executing multiple OT instances, our protocol allows a small saving, reducing the receiver communication complexity from 6 to 5 ring elements, but still achieving inverse logarithmic rate $1/O(\log n)$. So, if the number of parallel executions is very large (e.g., transferring $\Omega(n^3)$ bits), constant rate OT protcols would achieve better communication complexity than ours, by a logarithmic factor. However, this comes at a very high computational cost, as the amortization/compression only helps in reducing the communication complexity – the time and space (memory) complexity of those amortized protocols would be higher than ours by a much larger polynomial factor.

In summary, in a typical application setting, our protocol achieves much better communication and computational complexity than previous work. Communication is improved by at least a $O(n \log n)$ factor in the single execution setting, resulting in several orders of magnitude improvement in practice. Even when $n$ parallel executions are considered, we sill achieve at least $O(\log n)$ improvement in communication, and, in many cases, much more than that. When it comes to running time, our protocol outperforms previous work by a large marging both in theory and in practice. We remark that considering $n$ parallel OT executions only helps to reduce the communication complexity of previous protocols, and their running time still scales linearly (or worse, due to the overhead of compression/decompression) with the number of executions.

In our comparison, we focus on the communication complexity, as this parameter can be estimated in a way that is largely independent of the computational/implementation model, and a precise comparison can be carried out without the need to implement previous protocols, none of which have been implemented because clearly not practical. But it should be clear from our pseudocode, that our protocol would also be much faster than previous work, both asymptotically (by polynomial, typically quadratic $O(n^2)$ factors) and in practice (by several orders of magnitude.) See the next two sections for details.

### 4.1 Single Execution

The following table (Figure 5) compares asymptotic parameters, communication, and computational complexity for a single execution of the OT protocol. Much of the complexity of related protocols comes from the matrix multiplications that are required by key generation (in the case of [BD18] [DGI$^+$19] [BDGM19]) or by compression (in the case of [GH19]). So, we express the asymptotic complexity in terms of the matrix multiplication exponent $\omega \leq 3$. However, asymptotically faster matrix multiplication algorithms are likely to be only of theoretical interest, and for practical purposes, one should consider the value $\omega = 3$.

Our algorithm achieves quasi-constant $O(1/\log n)$ communication rate already in the single execution setting, improving other protocols by a superlinear $O(n \log n)$ factor. Some previous protocol [DGI$^+$19,BDGM19] achieve similar sender communication, but much higher communication from the receiver, which dominates the total communication cost.

The improvement in running time is even bigger. Our protocol essentially requires just a constant number of ring operations, which can be implemented (both in theory and in practice) in quasi-linear time $O(n \log n)$. The previous protocol achieving the best asymptotic complexity is that of [BD18], which has running time $O(n^\omega) > O(n^{2.3})$. This is already a substantial $\Omega(n^{1.3})$ theoretical improvement, But in practice, for $\omega = 3$, the improvement is almost quadratic $O(n^2)$, and with a protocol that is also arguably simpler and easier to implement. The other protocols are slower than ours by a quadratic factor $O(n^2)$ or worse. For typical values of the security parameter $n$ (in the hundreds) this is easily estimated to be a running time improvement by several orders of magnitude.

| Scheme | Modulus $q$ | Receiver Comm. (bits) | Sender Comm. (bits) | Overall Rate | Operations |
|---|---|---|---|---|---|
| [BD18] | $\Theta(n^3 \log^{2.5} n \cdot \gamma(n))$ | $\Theta(n^2 \log^2 n)$ | $\Theta(n \log^2 n)$ | $\Theta(1/n \log^2 n)$ | $\Theta(n^\omega)$ |
| [DGI$^+$19] | $\boldsymbol{\Theta(n^{2.5})}$ | $\Theta(n^2 \log^2 n)$ | $\boldsymbol{\Theta(n \log n)}$ | $\Theta(1/n \log^2 n)$ | $\Theta(n^3 \log n)$ |
| [GH19] | $\Omega(n^{17.5} \log^{10} n)$ | $\Theta(n^2 \log^2 n)$ | $\Theta(n^2 \log n)$ | $\Theta(1/n \log^2 n)$ | $\Omega(n^{1+\omega})$ |
| [BDGM19] | $\Theta(n^{2.5} \log^2 n)$ | $\Theta(n^2 \log^2 n)$ | $\boldsymbol{\Theta(n \log n)}$ | $\Theta(1/n \log^2 n)$ | $\Omega(n^3 \log^2 n)$ |
| This work | $\Theta(n^4 \gamma^6(n))$ | $\boldsymbol{\Theta(n \log n)}$ | $\boldsymbol{\Theta(n \log n)}$ | $\boldsymbol{\Theta(1/\log n)}$ | $\boldsymbol{\Theta(n \log n)}$ |

Fig. 5: **Comparison of Oblivious Transfer asymptotic parameters in the single execution setting.** Compared to prior work, our protocol reduces receiver communication by at least a factor $\mathcal{O}(n \log n)$, while matching the best prior sender communication. Our protocol also improves computational efficiency, requiring at least a factor $n$ fewer operations than prior work. The symbol $\omega$ above indicates the matrix multiplication constant, and $\gamma$ may be taken to be any function in $\omega(\sqrt{\log n})$. (The best parameters within each column are in bold face.)

To make the comparison more tangible, we propose a concrete setting of parameters achieving $\sim 120$ bits of security for the receiver, and compare to the statistically sender private OT protocols of [BD18], [DGI$^+$19], [BDGM19], and [GH19] with similar concrete security. (Security for the sender holds in a strong statistical sense, and can be easily estimated without making any computational assumption.) Following standard practice, the parameters of Table 6 were chosen based on the security estimates of the LWE security estimator [APS15].

Note that both [DGI$^+$19] and [BDGM19] have impressively low sender communication, due to rounding techniques that enable the receiver to correctly recover its chosen message given only some auxiliary information from the sender along with a single bit per bit of message. The concrete overall rate of these (and other prior) protocols is dominated by the receiver's communication though, and so the savings in download rate achievable by [DGI$^+$19] and [BDGM19] are lost when total communication is considered. On the other hand, our protocol's receiver communication is both asymptotically and concretely balanced with the sender's communication, giving an overall rate several orders of magnitude higher than prior work.

| Scheme | dim. $n$ | $\log q$ | Receiver Comm. (KB) | Sender Comm. (KB) | Msg. Length $\lvert m_b \rvert$ (KB) | Overall Rate |
|---|---|---|---|---|---|---|
| [BD18] | 900 | 40 | $3.24 \times 10^5$ | 190 | .113 | $1.5 \times 10^{-7}$ |
| [DGI+19] | 512 | 23 | 14000 | 2 | .064 | $4.6 \times 10^{-6}$ |
| [GH19] | 6800 | 255 | $3.8 \times 10^8$ | $1.5 \times 10^6$ | .85 | $2.2 \times 10^{-9}$ |
| [BDGM19] | 640 | 29 | 20000 | 2 | .08 | $4 \times 10^{-6}$ |
| This work | 2048 | 64 | 100 | 115 | .256 | $\mathbf{1.2 \times 10^{-3}}$ |

Fig. 6: **Concrete parameters achieving 120 bits of receiver security.** Compared to prior work, our protocol achieves the best overall rate by several orders of magnitude for a single execution of the protocol.

### 4.2 $O(n)$ Parallel Executions

Here we compare the parameters and efficiency of lattice-based SSP OT protocols for applications requiring $O(n)$ parallel executions of the protocol. In this setting, the compressibility of [GH19] can be utilized to obtain the same receiver and sender communication achieved in the single execution setting ($\Theta(n^2 \log^2 n)$), as the receiver can now pack encryptions of all $n$ of its choice bits into a single ciphertext, and all $n^2$ of the sender's bits may be similarly packed.

The compressibility of [BDGM19] is also now reflected in the sender communication. Their FHE scheme gives packed ciphertext lengths that are asymptotically $\max\{n \log q, \ell\}$, where $\ell$ is the total bit-length of the plaintext messages, and so the length of the plaintext messages dominates the sender communication in the parallel execution setting. However, the receiver is still required to send a large compression key comprising $n \log q$ encryptions with ciphertext size $n^2 \log q$, and so the overall rate of the OT protocol based on [BDGM19] will be dominated by this key.

Because we are considering $n$ parallel but independent executions of an OT protocol, rather than a single execution with large ($\mathsf{poly}(n)$) sender messages, the amortization required to achieve constant overall rate for the trapdoor hash function-based protocol of [DGI+19] is not possible. Similarly, our protocol and that of [BD18] require $\mathsf{poly}(n)$-length sender messages to achieve an improved amortized upload rate. For these protocols, the parameters and complexities given below (Figure 7) are simply those for running the base protocol $n$ times in parallel.

| Scheme | Modulus $q$ | Receiver Comm. (bits) | Sender Comm. (bits) | Overall Rate | Operations |
|---|---|---|---|---|---|
| [BD18] | $\Theta(n^3 \log^{2.5} n \cdot \gamma(n))$ | $\Theta(n^3 \log^2 n)$ | $\Theta(n^2 \log^2 n)$ | $\Theta(1/n \log^2 n)$ | $\Theta(n^{1+\omega})$ |
| [DGI+19] | $\boldsymbol{\Theta(n^{2.5})}$ | $\Theta(n^3 \log^2 n)$ | $\Theta(n^2 \log n)$ | $\Theta(1/n \log n)$ | $\Theta(n^5)$ |
| [GH19] | $\Omega(n^{27.5} \log^{15} n)$ | $\Theta(n^2 \log^2 n)$ | $\Theta(n^2 \log n)$ | $\Theta(1/ \log^2 n)$ | $\Omega(n^{2+\omega})$ |
| [BDGM19] | $\Theta(n^{4.5} \log^2 n)$ | $\Theta(n^3 \log^2 n)$ | $\boldsymbol{\Theta(n^2)}$ | $\Theta(1/n \log^2 n)$ | $\Omega(n^5 \log^2 n)$ |
| This work | $\Theta(n^4 \gamma^6(n))$ | $\boldsymbol{\Theta(n^2 \log n)}$ | $\Theta(n^2 \log n)$ | $\boldsymbol{\Theta(1/ \log n)}$ | $\boldsymbol{\Theta(n^2 \log n)}$ |

Fig. 7: **Comparison of asymptotic parameters.** Compared to prior work, our protocol improves in overall rate by at least a $\log n$ factor, and reduces the computational complexity by at least a factor $n$ for $n$ parallel executions of the SSP OT protocol. The symbol $\omega$ above indicates the matrix multiplication constant, and $\gamma$ may be taken to be any function in $\omega(\sqrt{\log n})$. (The best parameters in each column are in bold face.)

The last table (Figure 8) shows the concrete parameters for $n$ parallel executions of each OT protocol. Again we observe that the comparatively high upload rate of our protocol leads to a much better overall rate for applications requiring $n$ parallel OTs.

| Scheme | dim. $n$ | $\log q$ (bits) | Receiver Comm. (KB) | Sender Comm. (KB) | Msg. Length $\lvert m_b \rvert$ (KB) | Overall Rate |
|---|---|---|---|---|---|---|
| [BD18] | 900 | 40 | $2.92 \times 10^8$ | 190 | 102 | $1.5 \times 10^{-7}$ |
| [DGI+19] | 512 | 23 | $7.17 \times 10^6$ | 1024 | 33 | $4.6 \times 10^{-6}$ |
| [GH19] | 11000 | 1240 | $2.3 \times 10^{10}$ | $1.8 \times 10^7$ | 15125 | $6.6 \times 10^{-7}$ |
| [BDGM19] | 1300 | 54 | $8.0 \times 10^8$ | 220 | 211 | $2.6 \times 10^{-7}$ |
| This work | 2048 | 64 | 204800 | 235520 | 525 | **.0012** |

Fig. 8: **Concrete parameters achieving 120 bits of receiver security.** Compared to prior work, our protocol achieves the best overall rate by several orders of magnitude for $n$ parallel repetitions of the protocol.

## 5 Acknowledgements

We would like to thank Nicholas Genise and Daniel Kongsgaard for helpful conversations, and anonymous reviewers for useful suggestions.

## 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.

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. `http://eprint.iacr.org/2015/046`.

Ban93. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. 1993.

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.

BDGM19. Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Leveraging linear decryption: Rate-1 fully-homomorphic encryption and time-lock puzzles. In *TCC 2019, Part II*, LNCS, pages 407–437. Springer, Heidelberg, Germany, March 2019.

BGH13. Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed ciphertexts in LWE-based homomorphic encryption. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 1–13, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany.

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.

BGJ$^+$17. Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Dakshita Khurana, and Amit Sahai. Round optimal concurrent MPC via strong simulation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 743–775, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

BGJ$^+$18. Saikrishna Badrinarayanan, Vipul Goyal, Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Promise zero knowledge and its applications to round optimal MPC. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 459–487, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

CDLP14. Kai-Min Chung, Daniel Dadush, Feng-Hao Liu, and Chris Peikert. On the lattice smoothing parameter problem. *Proceedings of the Annual IEEE Conference on Computational Complexity*, 12 2014.

CF01. Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 19–40, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany.

DGH$^{+}$20. Nico Döttling, Sanjam Garg, Mohammad Hajiabadi, Daniel Masny, and Daniel Wichs. Two-round oblivious transfer from CDH or LPN. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 768–797. Springer, 2020.

DGI$^{+}$19. Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2019, Part III*, LNCS, pages 3–32, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

FP11. Lenny Fukshansky and Kathleen Petersen. On well-rounded ideal lattices, 2011.

GH19. Craig Gentry and Shai Halevi. Compressible FHE with applications to PIR. In *TCC 2019, Part II*, LNCS, pages 438–464. Springer, Heidelberg, Germany, March 2019.

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.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

Hay11. Masahito Hayashi. Exponential decreasing rate of leaked information in universal random privacy amplification. *IEEE Trans. Information Theory*, 57(6):3989–4001, 2011.

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

HLOV11. Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 70–88, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany.

ILL89. Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24, Seattle, WA, USA, May 15–17, 1989. ACM Press.

IPS09. Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Secure arithmetic computation with no honest majority. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 294–314. Springer, Heidelberg, Germany, March 15–17, 2009.

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 crytpography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 20–31, New York, NY, USA, 1988. ACM.

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.

KN08. Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 320–339, San Francisco, CA, USA, March 19–21, 2008. 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.

LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.

MR04. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press.

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.

OPP14. Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 536–553, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany.

PRS17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 461–473, Montreal, QC, Canada, June 19–23, 2017. ACM Press.

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany.

Rab05. Michael O. Rabin. How to exchange secrets with oblivious transfer, 2005. Harvard University Technical Report 81 talr@watson.ibm.com 12955 received 21 Jun 2005.

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.