New Communication-Efficient Oblivious Transfer Protocols Based on Pairings*

***** First public draft, April 17, 2007 *****

Helger Lipmaa

University College London, UK

Abstract. We construct two simple families of two-message $(n,1)$-oblivious transfer protocols based on degree-t homomorphic cryptosystems with communication of respectively $1+\lceil n/t \rceil$ and $3+\lceil n/(t+1) \rceil$ ciphertexts. The construction of both families relies on efficient cryptocomputable conditional disclosure of secret protocols. The currently most interesting case $t = 2$ can be based on the Boneh-Goh-Nissim cryptosystem. We use the new protocols to reduce the communication of virtually any existing oblivious transfer protocols by proposing a new communication-efficient generic transformation from computationally-private information retrieval protocols to oblivious transfer protocols.

Keywords. Computationally-private information retrieval, conditional disclosure of secrets, oblivious transfer.

1 Introduction

In an $(n,1)$-*oblivious transfer protocol for ℓ-bit strings,* $(n,1)$-*OT*$^\ell$, Alice on input $0 \leq \sigma \leq n-1$ retrieves the σth element of Bob's database $D = (D_0, \ldots, D_{n-1})$ where $D_i \in \{0,1\}^\ell$. One requires that Alice obtains no information about any D_j for $j \neq \sigma$, and that Bob obtains no information about σ. It is well-known that by general reductions, one can base both two-party computation and multi-party computation on $(2,1)$-OT. $(n,1)$-OT is a cornerstone of many handcrafted cryptographic protocols. Thus, it is important to construct efficient $(n,1)$-OT protocols that would be efficient for values of n ranging from $n = 2$ to say $n = 2^{20}$. The currently most communication-efficient $(n,1)$-OT protocols for large n were proposed in [Lip05,GR05], while some of the most communication-efficient $(2,1)$-OT protocols were proposed in [AIR01,LL07].

New linear protocols. We propose two new families $\mathsf{SimpleOT}_t$ and ExtOT_t, for $t \geq 1$, of linear-communication $(n,1)$-OT$^\ell$ protocols. Both families rely on a cryptosystem that enables to cryptocompute (i.e., compute-on-ciphertexts) degree-t polynomials with coefficients from $\mathbb{Z}_N \cup \{\star\}$ where $\star$ denotes a pseudorandom element of the plaintext group $\mathbb{Z}_N$. We call such a cryptosystem *degree-t homomorphic*. The case $t = 1$ includes additively homomorphic public-key cryptosystems like Paillier [Pai99], and the case $t = 2$ includes the BGN cryptosystem [BGN05].

* First draft, published only for the sake of early dissemination of the results.

Table 1. Comparison of different instantiations of ExtOT, SimpleOT with the protocols from [AIR01,LL07]. Here, k denotes the length of ciphertexts in bits; the values of $|\mathsf{pk}|$ and k depend on the underlying cryptosystem.

Protocol	Alice's comm.	Bob's comm.	Max ℓ	Cryptosystem	CDS eq.
Previous instantiations					
[AIR01] = $\mathsf{SimpleOT}_1$	$\lvert\mathsf{pk}\rvert + k$	nk	≤ 64	Mult. hom.	(6)
[LL07] = $\mathsf{SimpleOT}_1$	$\lvert\mathsf{pk}\rvert + k$	nk	≤ 430	Add. hom.	(6)
New instantiations					
$\mathsf{SimpleOT}_2$	$\lvert\mathsf{pk}\rvert + k$	$\lceil n/2\rceil k$	≤ 60	BGN	(6)
ExtOT_1	$\lvert\mathsf{pk}\rvert + 2k$	nk	≤ 430	Add. hom.	(5)
ExtOT_2	$\lvert\mathsf{pk}\rvert + 3k$	$\lceil n/3\rceil k$	≤ 60	BGN	(4)
Generic, hypothetical instantiations for $t > 2$					
$\mathsf{SimpleOT}_t$	$\lvert\mathsf{pk}\rvert + k$	$\lceil n/t\rceil k$	?	Hypothetical for $t > 2$	(6)
ExtOT_t	$\lvert\mathsf{pk}\rvert + 3k$	$\lceil n/(t+1)\rceil k$	?	Hypothetical for $t > 2$	(4)

Wlog, assume that $t \mid n$. Then, $(n,1)$-$\mathsf{SimpleOT}_t$ is a parallel repetition of n/t copies of an atomic $(t,1)$-$\mathsf{SimpleOT}_t$ protocol that use a common secret/public key pair. They also share Alice's first message that consists of the public key and of an encryption of Alice's index σ. In every single instance of $(t,1)$-$\mathsf{SimpleOT}_t$, Bob cryptocomputes his reply as a single encryption of the sum of two polynomials $\mathsf{cp}_i^{t-1}(\sigma)$ and $\mathsf{SimpleCDS}_i^t(\sigma)$, where the first polynomial takes care of the correctness and the second polynomial implements conditional disclosure of secrets (CDS, [AIR01,BGN05,LL07]) to guarantee Bob's privacy. More precisely, $\mathsf{cp}_i^t(\sigma)$ is the unique degree-t polynomial such that $\mathsf{cp}_i^t(\sigma) = D_\sigma$ if $\lfloor \sigma/t \rfloor = i$, and $\mathsf{SimpleCDS}_i^t(j)$ is a degree-t polynomial such that $\mathsf{SimpleCDS}_i^t(j) = 0$ for $\lfloor j/t \rfloor = i$ and $\mathsf{SimpleCDS}_i^t(j) = \star$ for $\lfloor j/t \rfloor \neq i$. Thus, $\mathsf{cp}_i^t(\sigma) + \mathsf{SimpleCDS}_i^t(\sigma)$ is equal to D_σ if $\lfloor \sigma/t \rfloor = i$, and to $\star$, otherwise. In particular, $\mathsf{SimpleOT}_1$ corresponds to the $(n,1)$-OT protocols from [AIR01,LL07].

The protocol $(n,1)$-ExtOT_t is similarly composed from atomic $(t+1,1)$-ExtOT_t protocols. Here, however, Bob's reply is a sum of $\mathsf{cp}_i^t(\sigma)$ and of a CDS polynomial $\mathsf{ExtCDS}'_i(\sigma)$ if $t = 1$, and of a CDS polynomial $\mathsf{ExtCDS}_i^t(\sigma)$ if $t > 1$. Because of the use of $\mathsf{cp}_i^t(\sigma)$, the number of atomic protocols is decreased to $\lceil n/(t+1)\rceil$. However, the corresponding CDS polynomials are more complicated and require Bob to communicate 2 ciphertexts per *atomic* protocol (if $t = 1$), or Alice to communicate 3 ciphertexts (if $t > 1$). The basic reason behind the added complexity is that there is no degree-t polynomial f such that $f(j) = 0$ for $\lfloor j/(t+1) \rfloor = i$ and $f(j) = \star$ for $\lfloor j/(t+1) \rfloor \neq i$.

Given the state of the art on existing degree-t homomorphic cryptosystems and efficient CDS protocols, one can instantiate the protocols $\mathsf{SimpleOT}_t$ and ExtOT_t with $t = 1$ or $t = 2$ as summarized in Table 1. Thus, the new protocols are communication-efficient even when n is small, say $n = 2$ or $n = 3$. See Sect. 3 for more comparison.

New sublinear protocols. The most communication-efficient known sublinear $(n,1)$-OT protocols are constructed by combining a communication-efficient CPIR protocol such as [Lip05,GR05] with a linear $(n,1)$-OT protocol from [AIR01,LL07].

The communication of the combined protocols is decreased when the protocols from [AIR01,LL07] are replaced with either $\mathsf{SimpleOT}_2$ or ExtOT_2. In the case of the only known CPIR protocol with log-communication [GR05], this replacement decreases slightly the communication of the combined protocol. In the case of the CPIR protocol from [Lip05], for small ℓ, the transformed oblivious transfer protocol is not only more secure but also more communication-efficient than Lipmaa's original CPIR protocol. We also point out that the existence of degree-2 cryptosystem with efficient decryption would imply the second log-communication oblivious transfer protocol.

Caveats. The proposed two-message protocols are secure only if the plaintext group order N has no small prime divisors. This means that if the group order is composite (like in the case of existing additively homomorphic cryptosystems or the BGN cryptosystem) then one can either rely on the PKI model or say use Lenstra's ECM algorithm to detect small divisors of N. See [LL07] for a discussion. This is not a problem if N is prime, e.g., if we rely on lifted Elgamal.

Notation. For a set S, $U(S)$ denotes the uniform distribution on it. Elements of the secret key are colored like this, elements of the public key are colored like this, (secret) plaintexts and randomizers are colored like this and ciphertexts are colored like this.

Road-map. In Sect. 2, we give necessary preliminaries. In Sect. 3, we describe the protocols $\mathsf{SimpleOT}_t$ and ExtOT_t. In Sect. 4, we describe a generic transformation of any $(n,1)$-CPIR protocol to a $(n,1)$-OT protocol with a comparable communication. In Sect. 5, we discuss related work.

2 Preliminaries

Bilinear groups. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of order N where $N = pq \in \mathbb{Z}$ and p, q are k-bit primes for some fixed security parameter $k \in \mathbb{Z}^+$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map, and for some fixed generator g of $\mathbb{G}$, $e(g,g)$ is a generator of $\mathbb{G}_T$. We assume that group operations and e are all efficiently computable. Let $\mathcal{G}$ be a bilinear group generation algorithm that outputs such a tuple $(p, q, \mathbb{G}, \mathbb{G}_T, e)$. [BGN05] suggest the following example. Pick large primes $p < q$ and let $N = pq$. Find the smallest ℓ so $P = \ell N - 1$ is prime and equal to 2 modulo 3. Consider the points on the elliptic curve $y^2 = x^3 + 1$ over $\mathbb{F}_P$. This curve has $P + 1 = \ell N$ points, so it has a subgroup $\mathbb{G}$ of order N. We let $\mathbb{G}_T$ be the order N subgroup of $\mathbb{F}^*_{P^2}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be the modified Weil pairing from [BF03]. We assume that k is a constant and work in the concrete security framework.

Public-key cryptosystems. A public-key cryptosystem is a tuple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ of algorithms with (possibly public-key dependent) plaintext space $\mathcal{M}$, randomizer space $\mathcal{R}$ and ciphertext space $\mathcal{C}$, such that $\mathcal{G}$ generates a random secret/public key pair $(\mathsf{sk}, \mathsf{pk})$, $\mathcal{E}_{\mathsf{pk}}(m; r) = c$ encrypts a plaintext $m \in \mathcal{M}$ to a ciphertext $c \in \mathcal{C}$ by using randomizer

$r \in \mathcal{R}$, and $\mathcal{D}_{\mathsf{sk}}(c) = m$ decrypts a ciphertext $c \in \mathcal{C}$ to a plaintext $m \in \mathcal{M}$. One requires that for any $(\mathsf{sk}, \mathsf{pk}) \in \mathcal{G}$ and for any $m \in \mathcal{M}, r \in \mathcal{R}$, $\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m; r)) = m$. A public-key cryptosystem is (τ, ε)-*IND-CPA secure* if for a freshly generated public/secret key pair $(\mathsf{sk}, \mathsf{pk})$, any τ-time adversary $\mathcal{A}$ can distinguish random encryptions of any two plaintext messages m_1, m_2, even chosen by himself, with probability $\leq \varepsilon$. (The probability is also taken over the choice of the keys.)

Additively homomorphic public-key cryptosystems. A public-key cryptosystem is *additively homomorphic* if $\mathcal{M} = (\mathbb{Z}_N, +, 0)$ for some integer N, $(\mathcal{C}, \cdot, 1)$ is a finite cyclic group, and if $\mathcal{D}_{\mathsf{sk}}(\mathcal{E}_{\mathsf{pk}}(m_1; r_1)\mathcal{E}_{\mathsf{pk}}(m_2; r_2)) = m_1 + m_2$ for any m_1, m_2, r_1, r_2. In addition, we require that $\mathcal{E}_{\mathsf{pk}}(m; r) \cdot \mathcal{E}_{\mathsf{pk}}(0; U(\mathcal{R})) = \mathcal{E}_{\mathsf{pk}}(m; U(\mathcal{R}))$ for any m, r; this enables to perform efficient rerandomization. There are many well-known additively homomorphic public-key cryptosystems, see e.g., [Pai99,DJ01].

Disclose-if-equal. For an additively homomorphic cryptosystem, given an encryption $c = \mathcal{E}_{\mathsf{pk}}(m; r)$ of some m, one can compute $c_1 \leftarrow c^{U(\mathbb{Z}_N)} h^{U(\mathcal{R})} = \mathcal{E}_{\mathsf{pk}}(m; r)^{U(\mathbb{Z}_N)} \cdot \mathcal{E}_{\mathsf{pk}}(0; U(\mathcal{R})) = \mathcal{E}_{\mathsf{pk}}(U(\mathbb{Z}_N) \cdot m; U(\mathcal{R}))$. If $\gcd(m, N) = 1$ (resp., $\gcd(m, N) > 1$) then $c_1 = \mathcal{E}_{\mathsf{pk}}(U(\mathbb{Z}_N); U(\mathcal{R}))$ is a random encryption of a random value from $\mathbb{Z}_N$ (resp., in some nontrivial subgroup of $\mathbb{Z}_N$). Laur and Lipmaa [LL07] defined a a protocol that forces c_1 to be an encryption of a (statistically) pseudorandom value of $\mathbb{Z}_N$ for *any* m. Briefly, in the implementation of the Laur-Lipmaa protocol, one redefines $\star$ (the "formal random element of $\mathbb{Z}_N$") to be the equal to

$$U(\mathcal{M}) + 2^\ell \cdot U(2^\ell \cdot \mathbb{Z}_{\lfloor N/2^\ell \rfloor}) \tag{1}$$

Because the distribution of $\star$ changes only slightly, we will still formally interpret $\star$ as being a uniformly random plaintext.

In a *disclose-if-equal* protocol, Alice on input a obtains Bob's input b_1 if $a = b_2$ for Bob's second input b_2, otherwise Alice obtains $\star$. In the Laur-Lipmaa disclose-if-equal protocol, given a random encryption of a, Bob computes a random encryption of $\star(b_2 - a) + b_1$. Alice recovers the answers modulo 2^ℓ with $\ell < p - 1 - \varepsilon$, where p is the smallest prime divisor of N and $2^{-\varepsilon}$ is the desired privacy level of honest Bob.

BGN cryptosystem and degree-t homomorphic cryptosystems. The BGN cryptosystem [BGN05] is defined as follows. The algorithm $\mathcal{K}$ runs $\mathcal{G}$ to generate $(p, q, \mathbb{G}, \mathbb{G}_T, e)$. Let $N \leftarrow pq$. Pick generators $g, u \leftarrow U(\mathbb{G})$ and let $h \leftarrow u^q$. Output public key $\mathsf{pk} \leftarrow (N, \mathbb{G}, \mathbb{G}_T, e, g, h)$ and private key $\mathsf{sk} \leftarrow p$. To encrypt a message $m \in \mathbb{Z}_{2^\ell}$ where $2^\ell < q$ with public key pk, pick a random $r \leftarrow \mathcal{R} := \mathbb{Z}_N$ and compute $\mathcal{E}_{\mathsf{pk}}(m; r) \leftarrow g^m h^r \in \mathbb{G}$. To decrypt a ciphertext c using the private key sk, compute first $c^p = (g^m h^r)^p = (g^p)^m$ and then recover m by computing the discrete logarithm of c^p on base g^p. This can be done in time $O(2^{\ell/2})$ and thus one must take say $\ell < 64$ or $\ell = O(\log k)$. Set $g_1 \leftarrow e(g, g)$ and $h_1 \leftarrow e(g, h)$ where g_1 has order N and h_1 has order q. Define associated cryptosystem $(\mathcal{E}^\top, \mathcal{D}^\top)$ in group $\mathbb{G}_T$, with $\mathcal{E}^\top_{\mathsf{pk}}(m; r) := g_1^m h_1^r$ where $\mathcal{D}^\top$ is defined as the discrete logarithm of $\mathcal{E}^\top_{\mathsf{pk}}(m; r)^p$ on base g_1^p.

Given BGN encryptions of any m_1, m_2, one can compute a BGN encryption of $m_1 + m_2$ as $\mathcal{E}_{pk}(m_1)\mathcal{E}_{\mathsf{pk}}(m_2)$, and an associated BGN encryption of $m_1 m_2$ as $e(\mathcal{E}_{\mathsf{pk}}(m_1), \mathcal{E}_{\mathsf{pk}}(m_2))$. In particular, $\mathcal{E}^{\top}_{\mathsf{pk}}(m; r) \equiv e(\mathcal{E}\mathsf{pk}(m; r), g)$. Thus, given BGN encryptions of any $m_1, \ldots, m_t$, one can compute associated BGN encryptions of

$$\mathcal{E}^{\top}_{\mathsf{pk}}(f(m_1, \ldots, m_t)) \tag{2}$$

for any quadratic polynomial $f \in (\mathbb{Z}_N \cup \{\star\})[M_1, \ldots, M_t]$. This generalizes the computations that one can do in the case of additively homomorphic cryptosystems, where the set of computable functions is restricted to linear polynomials $f \in (\mathbb{Z}_N \cup \{\star\})[M_1, \ldots, M_t]$. We call a cryptosystem *degree-t homomorphic* if one can similarly compute (associated) encryptions of type Eq. (2) for any degree-t polynomial $f \in (\mathbb{Z}_N \cup \{\star\})[M_1, \ldots, M_t]$.

In the case of the BGN cryptosystem one does *not* need interpret $\star$ as in Eq. (1) because of the inefficient decryption of the BGN: for Alice to be able to make use of incorrectly submitted inputs, she would need to be able to compute arbitrary discrete logarithms in the group of order p. However, this is just a byproduct of the inefficient decryption procedure of the BGN cryptosystem and thus should not be taken granted.

Conditional disclosure of secrets. During a conditional disclosure of secrets (CDS) protocol [AIR01,BGN05,LL07], Alice obtains Bob's secret exactly iff her own input belongs to some publicly specified set of valid inputs; if Alice's input is incorrect then Alice obtains usually a value that is statistically close to a uniformly random plaintext. There exist several general approaches of constructing CDS protocols that are cryptocomputable given a degree-t homomorphic cryptosystem. In particular, efficient cryptocomputable CDS protocols for many tasks, based on cryptosystems with $t = 1$ and $t = 2$, were respectively proposed in [AIR01,LL07] and [BGN05].

Oblivious transfer. Assume that Alice has an input $\sigma \in \{0, \ldots, n-1\}$ and Bob has a database $D = (D_0, \ldots, D_{n-1})$ where $D_i \in \{0,1\}^{\ell}$. In an $(n, 1)$-oblivious transfer protocol for ℓ-bit strings, $(n, 1)$-OT$^{\ell}$, Alice obtains D_σ and no additional information, and Bob obtains no information about σ. We only consider two-message oblivious transfer protocols. An oblivious transfer protocol is *correct* when in the case of honest parties, Alice receives D_σ. An oblivious transfer protocol is (τ, ε_1)-*private for Alice* if for any two indices σ_1, σ_2, even chosen by Bob himself, a τ-time Bob cannot distinguish the first messages of Alice that correspond to σ_1, σ_2. An oblivious transfer protocol is ε_2-*private for Bob* if there exists an unbounded simulator that, only given access to the first message of Alice and Bob's database element D_σ, generates Bob's second message from the distribution that is ε_2 close to Bob's response in the real protocol to Alice's first message. An oblivious transfer protocol is $(\tau, \varepsilon_1, \varepsilon_2)$-*relaxed-secure* if it is correct, (τ, ε_1)-private for Alice and ε_2-private for Bob. An (τ, ε_1)-secure $(n, 1)$-*computationally-private information retrieval (CPIR) protocol* is the same as a $(\tau, \varepsilon_1; 1)$-relaxed secure oblivious transfer protocol.

3 New Families of Oblivious Transfer Protocols

Assume that we have an underlying cryptosystem where one can compute degree-t polynomials on ciphertexts. We call such a cryptosystem *degree-t homomorphic*. For example, $t = 1$ in the case of additively homomorphic public-key cryptosystems and $t = 2$ in the case of the BGN cryptosystem. We next propose two families ExtOT_t and $\mathsf{SimpleOT}_t$ of linear-communication $(n, 1)$-OT protocols that use the properties of a degree-t cryptosystem to decrease the number of communicated ciphertexts to $3 + \lceil n/(t+1) \rceil$ and $1 + \lceil n/t \rceil$, respectively

Underlying idea of ExtOT_t. Wlog, assume that $(t + 1) \mid n$. The basic idea of the first new protocol, that we call $(n, 1)$-ExtOT, is simple. Alice first generates a new key pair for the degree-t homomorphic cryptosystem. She sends to Bob the new public key with a random encryption of σ. Given that, for every $0 \le i \le n/(t+1) - 1$, Bob cryptocomputes the polynomial $\mathsf{cp}_i^t(\sigma) + \mathsf{ExtCDS}_i^t(\sigma)$, where $\mathsf{cp}_i^t(\sigma)$ and $\mathsf{ExtCDS}_i^t(\sigma)$ are two degree-t polynomials that take care of protocols's correctness and Bob's privacy respectively. More precisely, cp_i^t is the unique degree-t polynomial, such that $\mathsf{cp}_i^t(\sigma) = D_\sigma$ if $\lfloor \sigma/(t+1) \rfloor = i$. For example,

$$\begin{aligned}
\mathsf{cp}_i^1(\sigma) =& D_{2i}((2i+1) - \sigma) + D_{2i+1}(\sigma - 2i) \ , \\
\mathsf{cp}_i^2(\sigma) =& D_{3i}((3i+1) - \sigma)((3i+2) - \sigma)/2 + D_{3i+1}(\sigma - 3i)((3i+2) - \sigma) + \\
& D_{3i+2}(\sigma - 3i)(\sigma - (3i+1))/2 \ .
\end{aligned}$$

Second, $\mathsf{ExtCDS}_i^t(\sigma)$ is a degree-t polynomial such that $\mathsf{ExtCDS}_i^t(\sigma) = 0$ if $\lfloor \sigma/(t+1) \rfloor = i$, and $\mathsf{ExtCDS}_i^t(\sigma) = \star$ otherwise. That is, ExtCDS_i^t implements a cryptocomputable conditional disclosure of secrets protocol. Therefore, $\mathsf{cp}_i^t(\sigma) + \mathsf{ExtCDS}_i^t(\sigma)$ is equal to D_σ if $\lceil \sigma/(t+1) = i \rceil$, and to $\star$ otherwise.

A minor complication here is that such polynomial would have degree $t + 1$. To overcome this issue, we let Alice to send to Bob three encryptions of $(\sigma_2, \sigma_1, \sigma_0)$, where

$$\sigma_2 \leftarrow \lfloor \sigma/(t+1) \rfloor \ , \sigma_1 \leftarrow \lfloor (\sigma \mod (t+1))/t \rfloor \ , \sigma_0 \leftarrow (\sigma \mod (t+1)) \mod t \ . \tag{3}$$

From these encryptions, Bob can cryptocompute an encryption of $\sigma = (t+1)\sigma_2 + t\sigma_1 + \sigma_0$. E.g., if $\sigma = 14$ and $t = 4$ then $\sigma_2 = 2$, $\sigma_1 = 1$, and $\sigma_1 = 0$. We now define

$$\mathsf{ExtCDS}_i^t(\sigma_2, \sigma_1, \sigma_0) := \star(\sigma_2 - i) + \star(\sigma_1 - 1)\sigma_1 + \star(\sigma_0 - t)(\sigma_0 - (t-1)) \cdots \sigma_0 + \star\sigma_1\sigma_0 \ . \tag{4}$$

Clearly, ExtCDS_i^t is a degree-t polynomial with the required properties, i.e., $\mathsf{ExtCDS}_i^t(\sigma_2, \sigma_1, \sigma_0) = 0$ if $\lceil \sigma/(t+1) \rceil = i$ and $\mathsf{ExtCDS}_i^t(\sigma_2, \sigma_1, \sigma_0) = \star$, otherwise. After that, Bob returns all $n/(t+1)$ ciphertexts to Alice who decrypts the $\lfloor \sigma/(t+1) \rfloor$th ciphertext and reduces the result modulo 2^ℓ. Thus, if $0 \le \sigma \le n-1$ then Alice retrieves D_σ, and if $\sigma \notin \{0, \ldots, n-1\}$ then Alice retrieves a close-to-uniformly random value.

The case $t = 1$ is different. In this case, we are not aware of a protocol with communication of $O(1) + \lceil n/2 \rceil$ ciphertexts. The main problem is that the CDS protocol for showing that $x \in \{0, 1\}$ by methods of [LL07] requires Bob to send *two* ciphertexts to

Alice, because there is no way to check that $\sigma_0 \in \{0,1\}$ by using a single linear polynomial. Instead, we transfer cp_i^1 twice, where the first time Alice obtains the answer if $\sigma_0 = 0$ and in the second time Alice obtains the answer if $\sigma_0 = 1$; this corresponds to the protocols of [AIR01,LL07]. More precisely, assume that $2 \mid n$. In ExtOT_1, Alice transfers to Bob one public key and two ciphertexts of $\sigma_1 = \lfloor \sigma/2 \rfloor$ and $\sigma_0 = \sigma \mod 2$. For every $0 \leq i \leq n/2 - 1$, Bob forwards to Alice random encryption of the vector $(\mathsf{cp}_i^1(\sigma), \mathsf{cp}_i^1(\sigma)) + \mathsf{ExtCDS}_i'(\sigma_1, \sigma_0)$, where

$$\mathsf{ExtCDS}_i'(\sigma_1, \sigma_0) := (\star \cdot (\sigma_1 - i) + \star \cdot \sigma_0, \star \cdot (\sigma_1 - i) + \star \cdot (\sigma_0 - 1)) \ . \qquad (5)$$

Thus, the communication of ExtOT_1 is 1 public-key and $2 + n$ ciphertexts.

Description of $(n, 1)$-ExtOT_2. We now follow up with a precise definition of the $(n, 1)$-ExtOT_t protocol. We only give an implementation in the case $t = 2$, i.e., when one uses the BGN cryptosystem. The general case is a straightforward extension.

Let $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the BGN cryptosystem. Assume Alice's private input is $0 \leq \sigma \leq n - 1$ and Bob's private input is $D = (D_0, \ldots, D_{n-1})$. Fix $\ell < \log_2 p$ such that doing $O(2^{\ell/2})$ steps is feasible; for example, $\ell := 64$.[1] Wlog, assume that $3 \mid n$. Denote by B the distribution $U(\mathbb{Z}_N) + 2^\ell \cdot U(\mathbb{Z}_{\lfloor N/2^\ell \rfloor})$. The protocol description follows:

1. Alice runs $\mathcal{K}$ to generate a new secret/public key pair $(\mathsf{sk}, \mathsf{pk})$. She stores sk. She computes $a_2 \leftarrow \mathcal{E}_{\mathsf{pk}}(\sigma_2; U(\mathcal{R}))$, $a_1 \leftarrow \mathcal{E}_{\mathsf{pk}}(\sigma_1; U(\mathcal{R}))$ and $a_0 \leftarrow \mathcal{E}_{\mathsf{pk}}(\sigma_0; U(\mathcal{R}))$, for σ_i computed according to Eq. (3), and sends $(\mathsf{pk}, a_2, a_1, a_0)$ to Bob.
2. If a_2, a_1 or a_0 is not a valid ciphertext then Bob rejects. Otherwise, Bob computes $a \leftarrow a_2^3 a_1^2 a_0$, and a vector of ciphertexts $\boldsymbol{b} = (b_1, \ldots, b_{n/3})$, where

$$\begin{aligned} g_i \leftarrow & e(a_2/\mathcal{E}_{\mathsf{pk}}(i;0), g)^B \cdot e(a_1/\mathcal{E}_{\mathsf{pk}}(1;0), a_1)^B \cdot e(a_0/\mathcal{E}_{\mathsf{pk}}(1;0), a_0)^B \cdot e(a_1, a_0)^B \ , \\ b_i \leftarrow & e(\mathcal{E}_{\mathsf{pk}}(3i-2;0)/a, \mathcal{E}_{\mathsf{pk}}(3i-1;0)/a)^{D_{3i}/2} \cdot \\ & e(a/\mathcal{E}_{\mathsf{pk}}(3i;0), \mathcal{E}_{\mathsf{pk}}(3i-2;0)/a)^{D_{3i-1}} \cdot \\ & e(a/\mathcal{E}_{\mathsf{pk}}(3i;0), a/\mathcal{E}_{\mathsf{pk}}(3i-1;0))^{D_{3i-2}/2} \cdot g_i \cdot h_1^{U(\mathcal{R})} \end{aligned}$$

   for $i \in \{1, \ldots, n/3\}$, and sends $\boldsymbol{b}$ to Alice.
3. Alice outputs $e \leftarrow \mathcal{D}_{\mathsf{pk}}^\top(b_{\lfloor \sigma/3 \rfloor}) \mod 2^\ell$.

Note that during his computation, Bob can several times reuse all the values of $\mathcal{E}_{\mathsf{pk}}(i;0)$, for $i \in \{0, \ldots, n/3\}$.

Theorem 1. *Assume that the BGN cryptosystem is (τ, ε_2)-IND-CPA secure, that the public key is correctly generated, and that $\ell < \log_2 p - \log_2 n - \varepsilon_2$ where $N = pq$ and $p < q$. Then the $(n, 1)$-ExtOT_2 protocol is $(\tau - O(1), 3\varepsilon_1; \varepsilon_2)$-relaxed-secure.*

Proof. CORRECTNESS: clearly, if a_j is generated correctly for $j \in \{0, 1, 2\}$, then b_i is a random associated encryption of a message distributed according to $X_i := \mathsf{cp}_i^2(\sigma) +$

[1] For the decryption to be polynomial-time in n, one needs that $\ell = O(\log n)$. However, in practical applications n is too small for the asymptotic notion to start to become relevant.

$\mathsf{ExtCDS}_i^2(\sigma_2, \sigma_1, \sigma_0)$. Clearly, if $\sigma = 3\sigma_2 + 2\sigma_1 + \sigma_0 \in \{3i, 3i+1, 3i+2\}$ then $e \equiv D_\sigma \pmod{2^\ell}$.

ALICE'S PRIVACY: the only thing Bob sees is 3 ciphertexts (together with a fresh public key pk). Therefore, Alice's privacy follows directly from the IND-CPA security of the BGN cryptosystem.

BOB'S PRIVACY: we need to construct a simulator that on inputs $(\mathsf{pk}, D_\sigma, a_2, a_1, a_0)$ solely, where pk is a random public key and $\sigma \leftarrow \mathcal{D}_{\mathsf{sk}}(a_2^3 a_1^2 a_0)$, computes a second round message that has almost the same distribution as b, i.e., that it is a random associated encryption of X_i. Simulator does the following. It rejects if any of a_i is not a valid ciphertexts. If $\sigma \notin \{0, \ldots, n-1\}$ then it outputs a random associated encryption of $U(\mathbb{Z}_N)$. Because of the results of [LL07], this is statistically ε_2-close to the random associated encryption of X_i. If $\sigma \in \{0, \ldots, n-1\}$ then the simulator outputs a random associated encryption of D_σ. Clearly, in this case simulator's output has distribution X_i. □

Note that $(n,1)$-ExtOT is secure only when one assumes that the public key is correctly generated. More precisely, one needs that the smallest prime divisor of N is sufficiently large, see [LL07]. This assumption can modeled by saying that this protocol is secure in the PKI model, or by letting Alice to prove once in zero knowledge that the public key is correct and then using the same public key in many instances of $(n,1)$-ExtOT. Yet another possibility is to use Lenstra's ECM algorithm to verify that N does not have small prime factors. These and other remedies are thoroughly discussed in [LL07].

Alternative family SimpleOT. We will next give a short description of the alternative family $\mathsf{SimpleOT}$ of $(n,1)$-OT^ℓ protocols. In $\mathsf{SimpleOT}_t$, Bob cryptocomputes polynomials $\mathsf{cp}_i^{t-1}(\sigma) + \mathsf{SimpleCDS}_i^t(\sigma)$, where cp_i^{t-1} is as defined before and $\mathsf{SimpleCDS}_i^t$ is another, simpler, CDS polynomial. More precisely, assume that $t \mid n$. In $\mathsf{SimpleOT}_t$, Alice transfers a new public key and a random encryption of σ, and Bob replies with n/t random encryptions of $\mathsf{cp}_i^t(\sigma) + \mathsf{SimpleCDS}_i^t(\sigma)$, where

$$\mathsf{SimpleCDS}_i^t(\sigma) := \star(\sigma - ti) \cdot \cdots \cdot (\sigma - (ti + t - 1)) \tag{6}$$

for $0 \le i \le n/t - 1$.

Therefore, in $\mathsf{SimpleOT}_t$, Alice transfers 1 public key and 1 ciphertext, while Bob transfers $\lceil n/t \rceil$ ciphertexts (as opposed to 3 and $\lceil n/(t+1) \rceil$ ciphertexts in the case of ExtOT). Clearly, $\mathsf{SimpleOT}_1$ corresponds to the oblivious transfer protocol from [AIR01,LL07]. The only other current instantiation is $\mathsf{SimpleOT}_2$ when coupled with the BGN cryptosystem. To the best of our knowledge, if $\ell \le 64$, $\mathsf{SimpleOT}_2$ is the most communication-efficient available $(2,1)$-OT^ℓ protocol, having the total communication of 1 public key and 2 ciphertexts.

Comparison. In the case $t = 1$, the underlying cryptosystem must be additively homomorphic. One can use either the lifted Elgamal (that has inefficient decryption) or say the Paillier [Pai99] or the Damgård-Jurik [DJ01]. Then, $\mathsf{SimpleOT}_1$ corresponds resp. to the Aiello-Ishai-Reingold protocol [AIR01] or to the Laur and Lipmaa protocol [LL07], while ExtOT_1 is a related but slightly less efficient protocol. Compared to

the case $t = 2$, the case $t = 1$ benefits from the existence of a wide variety of additively homomorphic public-key cryptosystems, shorter public keys, and efficient decryption that makes it possible to obliviously transfer long strings with say $\ell \geq 400$. On the other hand, the number of transferred ciphertexts is larger than in the case of $t = 2$. Moreover, the ciphertexts of existing additively homomorphic cryptosystems are twice longer than the ciphertexts of the BGN cryptosystem. On the other hand, the ciphertexts of lifted Elgamal are shorter than the ciphertexts of the BGN cryptosystem.

In the case of $t = 2$, one uses a degree-2 homomorphic cryptosystem, e.g., the Boneh-Goh-Nissim cryptosystem [BGN05]. Compared to $t = 1$, one now transfers less ciphertexts. Additionally, because these instantiations operate on the ciphertexts of the BGN cryptosystem, they can be used in conjunction with other protocols that rely on the BGN cryptosystem; such applications include efficient non-interactive zero-knowledge proofs from [GOS06]. On the other hand, one is currently restricted to the BGN cryptosystem that has longer public keys, compared to existing additively homomorphic public-key cryptosystems, and inefficient decryption that only allows to efficiently transfer strings with say $\ell \leq 64$.

From the communication-efficiency view-point, when neglecting the length of the public key and assuming that ℓ is small, for $n \leq 15$, the most efficient new protocol is $(n, 1)$-$\mathsf{SimpleOT}_2$, while for $n > 15$, the most efficient protocol is $(n, 1)$-ExtOT_2. Note that in many common applications of oblivious transfer, the public key is shared with other protocols and thus does not incur a communication overhead.

4 Sublinear Oblivious Transfer

A common methodology to construct $(n, 1)$-OT protocols is to first construct a communication-efficient $(n, 1)$-CPIR protocol and then apply an efficient transformation to transfer it to a comparably efficient $(n, 1)$-OT protocol. Examples of communication-efficient $(n, 1)$-CPIR protocols include [Lip05,GR05]. A typical transformation was proposed in [AIR01] and later refined in [LL07] to work with existing additively homomorphic cryptosystems. Next, we generalize the approach of [AIR01,LL07].

We now describe a new transformation based on ExtOT_t for $t > 1$; the transformation based on $\mathsf{SimpleOT}_t$ is similar. Wlog, assume that $(t+1) \mid n$. Recall that during the ExtOT_t protocol, Bob first constructs a database of $n/(t+1)$ ciphertexts, such that the ith ciphertext encrypts D_σ if $\lfloor \sigma/(t+1) \rfloor = i$, and $\star$, otherwise. Then Bob transfers the whole database of ciphertexts to Alice. Instead, we can use in parallel *any* two-message $(n/(t+1), 1)$-CPIR protocol so that Alice will only obtain the $\lfloor \sigma/(t+1) \rfloor$th ciphertext (or possibly more). The resulting transformed protocol is clearly relaxed-secure: first, because ExtOT_t is relaxed-secure even if Alice sees *all* intermediate ciphertexts, the composed protocol is also relaxed-secure. Second, Bob only sees the first messages of Alice of both protocols and thus the composed protocols preserves Alice's privacy iff both ExtOT_t and the used CPIR protocol preserve Alice's privacy.

In general, let Π_1 be the ExtOT_t (or say the $\mathsf{SimpleOT}_t$) protocol, and let Π_2 be an arbitrary CPIR protocol. We denote the transformed protocol by $\Pi_2 \circ \Pi_1$, the case $\Pi_1 = \mathsf{SimpleOT}_1$ corresponds to the transformation proposed in [AIR01,LL07].

Clearly, if Π_1 on database elements of length ℓ has the first message of $C_1(n,\ell)$ bits and the second message of $C_2(n,\ell)$ ciphertexts, and Π_2 on database elements of length k has communication of $C_3(n,k)$ bits, then the transformed protocol $\Pi_2 \circ \Pi_1$ has communication of $C_1(n,\ell) + C_3(C_2(n,\ell),k)$ bits. Here, k is the length of ciphertexts in bits. Thus, $\Pi_2 \circ \mathsf{SimpleOT}_1$ has communication of $|\mathsf{pk}| + \lceil 2\log_2 N \rceil + C_3(n, \lceil 2\log_2 N \rceil)$, where $|\mathsf{pk}| = \lceil \log_2 N \rceil \approx 1024$ bits. On the other hand, $\Pi_2 \circ \mathsf{ExtOT}_t$ has communication of $|\mathsf{pk}| + 3\lceil \log_2 N \rceil + C_3(\lceil n/(t+1) \rceil, \lceil \log_2 N \rceil)$, where $|\mathsf{pk}|$ is somewhat longer compared to the case of $\mathsf{SimpleOT}_1$.

If Π_2 is the Gentry-Ramzan CPIR protocol [GR05] with communication $O(\log_2 n + \ell)$ then the total communication of $\Pi_2 \circ \mathsf{SimpleOT}_1$ is $|\mathsf{pk}| + O(\log_2 n + 2\log_2 N)$. In this case, the total communication of $\Pi_2 \circ \mathsf{ExtOT}_t$ is not significantly different unless t is large. On the other hand, the communication decrease is significant in the case of less communication-efficient CPIR protocols. Recall that Lipmaa's $(n,1)$-CPIR protocol [Lip05]—when used on top of the Damgård-Jurik cryptosystem [DJ01]—has communication of $(\frac{1}{2} \cdot \log_2^2 n + (s+3/2) \cdot \log_2 n + s)k$ bits, where $k = \lceil \log_2 N \rceil$. Thus, applying Lipmaa's CPIR protocol on the ExtOT_2-transformed database of $n/3$ ciphertexts results in the protocol $\Pi_2 \cdot \mathsf{ExtOT}_2$ that has communication complexity $(3(s+1) + \frac{1}{2} \cdot \log_2^2(n/3) + ((s+1)+3/2) \cdot \log_2(n/3) + (s+1))k = (\frac{1}{2}\log_2^2 n + (s + \frac{5}{2} - \log_2 3)\log_2 n + (4 - \log_2 3)s + 4 + 5/2 \cdot \log_2 3 + \frac{1}{2} \cdot \log_2^2 3)k$. This means that—assuming that the strings to be transferred are short—the ExtOT_2-transformation actually *reduces* the communication of Lipmaa's original CPIR protocol, on top of increasing its security.

Recursive ExtOT_t. We can recursively apply ExtOT_t to itself. Bob's original database has n items, each of ℓ bits. intermediate database, generated by ExtOT_t has $\lceil n/(t+1) \rceil$ ciphertexts, each of $\lceil \log_2 N \rceil$ bits. One can next apply the $(\lceil n/(t+1) \rceil, 1)$-$\mathsf{ExtOT}_t$ protocol $\xi := \lceil \log_2 N/\ell \rceil$ times to retrieve all $\lceil \log 2N \rceil$ bits of the $\lceil n/(t+1) \rceil$th intermediate ciphertext. Continuing, in the level r recursion, Alice sends 1 public key and $3r$ ciphertexts and Bob sends $\xi^{r-1} \cdot \lceil n/(t+1)^{r-1} \rceil$ ciphertexts.

Interestingly, if there existed a degree-2 homomorphic cryptosystem with $\xi = 2$ then this recursive construction would result in a $O(\log n)$ communication $(n,1)$-OT protocol. More precisely, $r \leftarrow (\ln n - \ln 6 + \ln\ln 1.5)/\ln 1.5$ would result in the optimal communication of $(3\ln n + 3 - 3\ln 6 + 3\ln\ln 1.5)/\ln 1.5 \approx 5.1\log_2 n - 12.5$ ciphertexts. The same asymptotic result holds whenever $\xi < t$, while the optimal case for $\xi \geq t$ is just the trivial one with $r = 1$.

5 Related Work

Boneh, Goh and Nissim [BGN05] considered the application of degree-2 homomorphic cryptosystems to construct efficient oblivious transfer protocols. They proposed two essentially different $(n,1)$-OT protocols. Both protocols handle D as a two-dimensional square D_{ij}. Alice's query is a pair of coordinates (σ_1, σ_2) to this square. The first OT protocol assumes that $\ell = 1$ and requires communication $O(\sqrt{n} \cdot k)$, where k is the security parameter (the length of ciphertexts in bits). Here, Alice sends $2\sqrt{n}$ encryptions

of Boolean values x_i, y_j, where $x_{\sigma_1} = y_{\sigma_2} = 1$ and $x_i = y_j = 0$, otherwise. Bob cryptocomputes the formula $\vee_{D_{ij}}(x_i \wedge y_j)$.

In the second OT protocol, $\ell = O(\log n)$ as in $(n,1)$-ExtOT. Alice generates two polynomials p_1, p_2 such that $p_i(a) = 1$ for $a = \sigma_i$ and $p_i(a) = 0$ for other values of $a \in [\sqrt{n}]$, and sends their coordinates to Bob. Bob cryptocomputes the encryption of $\sum_{i,j} p_1(i)p_2(j)D_{ij}$. Alice recovers from it D_{σ_1,σ_2} in time $O(2^{\ell/2}) = O(n^{O(1)})$.

Finally, one uses communication-balancing techniques to lower the communication. The database is viewed as comprising of $n^{1/3}$ chunks, each chunk containing $n^{2/3}$ entires, where Alice is interested in retrieving entry (I,J,K) of D. Alice sends Bob the coefficients of two polynomials $p_1(x)$ and $p_2(x)$ of degree $\sqrt[3]{n} - 1$ such that $p_1(i) = p_2(i) = 0$ on $0 \le i < \sqrt[3]{n}$ except for $p_1(I) = p_2(J) = 1$. Bob uses the encryption scheme's homomorphic properties to compute encryptions of $D_{I,J,k} = \sum_{0 \le i,j < \sqrt[3]{n}} p_1(i)p_2(j)D_{i,j,k}$ for $0 \le k < \sqrt[3]{n}$. Bob sends the $\sqrt[3]{n}$ resulting ciphertexts to Alice who decrypts the Kth entry. Recursively applying this scheme results in a communication complexity $O(n^{\varepsilon}k)$ for any $\varepsilon > k$ [BGN05].

The essential differences, compared to our solutions, are: $(n,1)$-ExtOT_2 requires Alice to send three ciphertexts and Bob to send $\lceil n/3 \rceil$ ciphertexts, while the protocols of [BGN05] that correspond to one-dimensional case require Alice to send n ciphertexts and Bob to send one ciphertext. This means that $(n,1)$-ExtOT_2 is approximately 3 times more communication-efficient than the one-dimensional BGN protocols. Moreover, one can combine ExtOT_t and $\mathsf{SimpleOT}_t$ with an arbitrary existing sublinear computationally-private information retrieval protocol to construct an almost as efficient oblivious transfer protocol. The oblivious transfer protocols from [BGN05] do not seem to share this property. In the case of protocols of [BGN05] it seems that one can only use standard communication-balancing techniques that are not in par with the state-of-the art methods of [Lip05,GR05].

Open problems and acknowledgments. Constructing a degree-2 homomorphic cryptosystem with efficient decryption is a major open problem. As we showed in Sect. 4, such a cryptosystem would make it possible to construct another $(n,1)$-OT protocol with $O(\log n)$ communication. Constructing degree-t, for $t > 2$, homomorphic cryptosystems is another well-known open problem. A more specific open problem posed by this paper is to construct a degree-1 homomorphic cryptosystem based $(n,1)$-OT protocol (e.g., a more efficient version of ExtOT_1) with communication $O(1) + n/2$.

We would like to thank Jens Groth for helpful comments. This paper was started while the author was visiting the Chinese University of Hong Kong, we would like to thank Victor K. Wei for generous support.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced Oblivious Transfer: How to Sell Digital Goods. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In Joe Kilian, editor, *The Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, Cambridge, MA, USA, February 10–12, 2005. Springer Verlag.

[DJ01] Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *Public Key Cryptography 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136, Cheju Island, Korea, February 13–15, 2001. Springer-Verlag.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect Non-Interactive Zero-Knowledge for NP. In Serge Vaudenay, editor, *Advances in Cryptology — EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 338–359, St. Petersburg, Russia, May 28–June 1, 2006. Springer-Verlag.

[GR05] Craig Gentry and Zulfikar Ramzan. Single-Database Private Information Retrieval with Constant Communication Rate. In Luis Caires, Guiseppe F. Italiano, Luis Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *The 32nd International Colloquium on Automata, Languages and Programming, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 803–815, Lisboa, Portugal, 2005. Springer-Verlag.

[Lip05] Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Communication. In Jianying Zhou and Javier Lopez, editors, *The 8th Information Security Conference (ISC'05)*, volume 3650 of *Lecture Notes in Computer Science*, pages 314–328, Singapore, September 20–23, 2005. Springer-Verlag.

[LL07] Sven Laur and Helger Lipmaa. A New Protocol for Conditional Disclosure of Secrets And Its Applications. In Jonathan Katz and Moti Yung, editors, *5th International Conference on Applied Cryptography and Network Security – ACNS'07*, volume 4521 of *Lecture Notes in Computer Science*, pages 207–225, Zhuhai, China, June 5–8, 2007. Springer-Verlag.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, May 2–6, 1999. Springer-Verlag.