Single Database Private Information Retrieval with Logarithmic Communication

Yan-Cheng Chang
Harvard University
ycchang@eecs.harvard.edu

February 10, 2004

Abstract

In this paper, we study the problem of single database private information retrieval, and present schemes with only logarithmic server-side communication complexity. Previously the best result could only achieve polylogarithmic communication, and was based on certain less well-studied assumptions in number theory [CMS99]. On the contrary, our construction is based on Paillier's cryptosystem [P99], which along with its variants have drawn extensive studies in recent cryptographic researches [PP99, G00, CGGN01, DJ01, CGG02, CNS02, ST02, GMMV03, KT03], and have many important applications (e.g., the Cramer-Shoup CCA2 encryption scheme in the standard model [CS02]).

Actually, our schemes can be directly used to implement 1-out-of-N *ℓ-bit string* oblivious transfer with $O(\ell)$ sender-side communication complexity (against semi-honest receivers and malicious senders). Note the sender-side communication complexity is independent of N, the constant hidden in the big-O notation is in fact small, and ℓ is unrestricted. Moreover, We also show a way to do communication balancing between the sender-side and the receiver-side.

In addition, we show how to handle malicious receivers with small communication overheads, which itself is a non-trivial result.

1 Introduction

Single database private information retrieval (1dPIR) is a cryptographic protocol between a database server, who has an N-bit database x, and a user, who has an index $1 \leq i \leq N$, such that the user can learn the i-th bit of x without revealing his index while the database server can send less than N bits to the user (as otherwise the problem becomes trivial). In addition to its numerous applications [A01], 1dPIR is a very strong cryptographic primitive in the sense that it can be used to construct oblivious transfer [CMO00], a cryptographic primitive that is known to be *complete* for secure computations [K88]. Historically, the first 1dPIR scheme was proposed in [KO97], with its security based on the hardness of the quadratic residuosity problem and with $O(N^\epsilon)$ server-side communication complexity for any constant ϵ. After that, in fact, only a few implementations of 1dPIR were found.

Specifically, a scheme with polylogarithmic server-side communication complexity was proposed in [CMS99]; however, its security was based on some less well-studied assumptions in number theory, i.e. the hardness of Φ-Hiding and the existence of Φ-Sampling. Besides, there is a result showing that 1dPIR can be built using trapdoor permutations [KO00]. But since the result of [KO00] is reduction-oriented, it actually requires more server-side communication than the previous ones.

In this paper, we present schemes for $\mathsf{1dPIR}$ with only logarithmic server-side communication complexity, which break the polylogarithmic bound given in [CMS99]. Notably, we also have less communication from the user-side. Our schemes are based on the additive homomorphic properties of Paillier's cryptosystem [P99], which is *semantically secure* under the Composite Residuosity Assumption (CRA). CRA is the extension of the well-studied Quadratic Residuosity Assumption (QRA) stating that it is computationally intractable to decide whether a random element in $\mathbb{Z}_n^*$ has a square root modulo n, where n is a RSA modulus. And CRA states that it is computationally intractable to decide whether a random element in $\mathbb{Z}_{n^2}^*$ has an n-th root modulo n^2.

Because Paillier's cryptosystem along with its variants have drawn extensive studies in recent cryptographic researches [PP99, G00, CGGN01, DJ01, CGG02, CNS02, ST02, GMMV03, KT03], and have many important applications (e.g., the Cramer-Shoup CCA2 encryption scheme in the standard model [CS02]), we believe CRA could be a good candidate for hardness assumption.

Supposing the security parameter is $O(\log N)$ bits in length, we can use the following table to compare our result with other known $\mathsf{1dPIR}$ schemes: (Here $d \gg 1$, and $\epsilon < 1$ can be any constant.)

	Server-side Comm.	User-side Comm.	Computational Assumption
[KO97]	$O(N^\epsilon)$	$O(N^\epsilon \log N)$	Quadratic Residuosity is hard
[CMS99]	$O((\log N)^d)$	$O((\log N)^4)$	Φ-Hiding is hard, $\exists$ Φ-Sampling
[KO00]	$N(1 - \frac{1}{6N^\epsilon}) + O(N^{2\epsilon})$	$O(N^{2\epsilon})$	$\exists$ Trapdoor Permutations
Our result	$O(\log N)$	$O(N^\epsilon \log N)$	Composite Residuosity is hard

Clearly, our result beats all previous solutions regarding the server-side communication complexity. In fact, our schemes can be directly used to implement 1-out-of-N *ℓ-bit string* oblivious transfer ($\binom{N}{1}\mathsf{OT}^\ell$), which is a cryptographic protocol between a sender, who has N ℓ-bit strings, and a receiver, who has an index $1 \le i \le N$, such that receiver can obtain the i-th string from sender without revealing his index and can learn nothing more. Our implementation of $\binom{N}{1}\mathsf{OT}^\ell$ only requires $O(\ell)$ sender-side communication complexity, and is secure against semi-honest receivers and malicious senders. Note the sender-side communication complexity is independent of N, the constant hidden in the big-O notation is in fact small, and ℓ is unrestricted.

Moreover, we show a way to do communication balancing between the sender-side and the receiver-side, and show a way to make our implementation secure against malicious receivers under CRA with small communication overheads. We emphasize that the later result is non-trivial.

We organize the rest of this paper as follows. In section 2, we first define $\mathsf{1dPIR}$ and $\binom{N}{1}\mathsf{OT}^\ell$ and then introduce CRA as well as properties of Paillier's cryptosystem. In section 3, we give several schemes for $\mathsf{1dPIR}$ with communication-efficiency of different levels, and show how to use them to implement efficient schemes for $\binom{N}{1}\mathsf{OT}^\ell$ with capability of doing communication balancing. In Section 4, finally, we consider the case of malicious receivers.

2 Preliminaries

For an integer $\ell \in \mathbb{N}$, let $[\ell]$ denote the set $\{1, 2, \cdots, \ell\}$. For an N-bit string x, let $x[i]_{i \in [N]}$ denote its i-th bit. A *semi-honest* player always follows the protocol properly with the exception that it keeps a record of all its intermediate computations [G98]. On the other hand, we put no restriction on the behavior of a *malicious* player. We use the notation $a \xleftarrow{R} A$ to denote choosing an element a uniformly at random from the set A, and use PPT to denote *probabilistic polynomial time.* Also, we say a function is negligible in k if for any polynomial p there exists a k_0 such that for all $k > k_0$ we have $f(k) < 1/p(k)$. All logarithms in this paper have base 2.

Moreover, an encryption scheme is *semantically secure* if it hides all partial information of the input, or equivalently, if it is *polynomial time indistinguishable*, i.e. there is no adversary can find even two messages which encryptions he can distinguish between [GM84]. We state them formally as follows.

Definition 1. *A probabilistic encryption scheme $\mathcal{E}$ with security parameter k, input domain $\mathcal{M}(k)$ and randomness domain $\mathcal{R}(k)$ is said to be* semantically secure *if for any PPT algorithm A, any message $m \in \mathcal{M}(k)$ and any function h, there is PPT algorithm B such that the following value is negligible in k:*

$$|\mathbf{Pr}[A(1^k, c) = h(m) | \ r \stackrel{R}{\leftarrow} \mathcal{R}(k),\ c = \mathcal{E}(m, r)] - \mathbf{Pr}[B(1^k) = h(m)]|.$$

Definition 2. *A probabilistic encryption scheme $\mathcal{E}$ with security parameter k, input domain $\mathcal{M}(k)$ and randomness domain $\mathcal{R}(k)$ is said to be* polynomial time indistinguishable *if for any PPT algorithm A and any two messages $m_0, m_1 \in \mathcal{M}(k)$, the following value is negligible in k:*

$$|\mathbf{Pr}[A(1^k, m_0, m_1, c) = m_b | \ b \stackrel{R}{\leftarrow} \{0,1\},\ r \stackrel{R}{\leftarrow} \mathcal{R}(k),\ c = \mathcal{E}(m_b, r)] \ - \ 1/2|.$$

Lemma 1. *[GM84] A probabilistic encryption scheme is semantically secure if and only if it is polynomial time indistinguishable.*

2.1 Single database private information retrieval and oblivious transfer

In this section, we define $\mathsf{1dPIR}$ and t-out-of-N ℓ-bit string oblivious transfer ($\binom{N}{t}\mathsf{OT}^\ell$).

Definition 3. Single database private information retrieval ($\mathsf{1dPIR}$) *is a protocol between two players* Server*, who has an N-bit string x, and* User*, who has an index $i \in [N]$, that guarantees*

1. *Correctness: User can learn $x[i]$ and Server can send less than N bits to User, and*
2. *User's security: for any PPT algorithm A and any $j \in [N]$, the following value is negligible in the security parameter k:*
$$|\mathbf{Pr}[A(1^k, C_k(i)) = 1] - \mathbf{Pr}[A(1^k, C_k(j)) = 1]|,$$
where $C_k(y)$ is the distribution of communication from User induced by an index $y \in [N]$.

Definition 4. t-out-of-N ℓ-bit string oblivious transfer ($\binom{N}{t}\mathsf{OT}^\ell$) *is a protocol between two players* Sender*, who has N ℓ-bit strings $x_1, x_2, \cdots, x_N$, and* Receiver*, who has t indexes $i_1, i_2, \cdots, i_t \in [N]$, that guarantees*

1. *Correctness: User can learn $x_{i_1}, x_{i_2}, \cdots, x_{i_t}$, and*
2. *Receiver's security: for any PPT algorithm A and any $j_1, j_2, \cdots, j_t \in [N]$, the following value is negligible in the security parameter k:*
$$|\mathbf{Pr}[A(1^k, C_k(i_1, i_2, \cdots, i_t)) = 1] - \mathbf{Pr}[A(1^k, C_k(j_1, j_2, \cdots, j_t)) = 1]|,$$
where $C_k(y_1, y_2, \cdots, y_t)$ is the distribution of communication from Receiver induced by indexes $y_1, y_2, \cdots, y_t \in [N]$, and
3. *Sender's security: for any PPT algorithm A and any $x'_1, x'_2, \cdots, x'_N \in \{0,1\}^\ell$ such that $x'_{i_1} = x_{i_1}, x'_{i_2} = x_{i_2}, \cdots, x'_{i_t} = x_{i_t}$, the following value is negligible in the security parameter k:*
$$|\mathbf{Pr}[A(1^k, C_k(x_1, x_2, \cdots, x_N)) = 1] - \mathbf{Pr}[A(1^k, C_k(x'_1, x'_2, \cdots, x'_N)) = 1]|,$$
where $C_k(z_1, z_2, \cdots, z_N)$ is the distribution of communication from Sender induced by strings $z_1, z_2, \cdots, z_N \in \{0,1\}^\ell$.

2.2 Composite residuosity assumption

Let $n = pq$ be a RSA modulus, i.e. product of two safe primes of the same length in bits. (A prime p is *safe* if it has the form of $2q + 1$ with q also a prime). Consider the multiplicative group $\mathbb{Z}_{n^2}^*$.

Definition 5. *An element $z \in \mathbb{Z}_{n^2}^*$ is said to be an n-th residue if there exists an element $y \in \mathbb{Z}_{n^2}^*$ such that $z = y^n \bmod n^2$, otherwise it is said to be an n-th non-residue.*

Note the problem to distinguish n-th residues from n-th non-residues, like the problem to decide quadratic residues and quadratic non-residues, is *random-self-reducible*, i.e. each instance of the problem is an average case [P99]. Specifically, all instances of a random-self-reducible problem are either uniformly intractable or uniformly solvable in polynomial time [BM84].

Definition 6. *Composite Residuosity Assumption (CRA): If the factorization of n is unknown, there is no PPT distinguisher for n-th residues modulo n^2 [P99].*[1]

Note due to the random-self-reducibility, the validity of CRA only depends on the choice of n [P99].

2.3 Paillier's cryptosystem

Let $n = pq$ be a RSA modulus, i.e. product of two safe primes of the same length in bits. Consider the multiplicative group $\mathbb{Z}_{n^2}^*$. Given any $g \in \mathbb{Z}_{n^2}^*$ whose order is a non-zero multiple of n (for example, $g = n + 1$), it can be shown that g induces a bijection [P99]:

$$\begin{array}{rcl} \mathcal{E}_g(a, b) & = & g^a b^n \bmod n^2. \\ (\mathbb{Z}_n \times \mathbb{Z}_n^* & \rightarrow & \mathbb{Z}_{n^2}^*) \end{array}$$

In other words, for every element $w \in \mathbb{Z}_{n^2}^*$, there exists a unique pair $(a, b) \in \mathbb{Z}_n \times \mathbb{Z}_n^*$ such that we have $w = g^a b^n \bmod n^2$, and vice versa. We know under CRA it is computationally intractable to compute a given only w, n and g, as otherwise we can decide the n-th residuosity of w. However, if we know the factorization of n, we can compute a using the following method [P99]:

$$a = \mathcal{D}_g(w) = \frac{L(w^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n,$$

where $L(u) = (u - 1)/n$ for $u \in \mathbb{Z}_{n^2}^*$, and $\lambda = lcm(p - 1, q - 1)$.

Accordingly, Paillier defines a probabilistic public-key cryptosystem using $\mathcal{E}_g$ as the encryption scheme for any message $a \in \mathbb{Z}_n$ with randomness $b \in \mathbb{Z}_n^*$ [P99]. Specifically, the public keys are n and g, the private key is the factorization of n, and $\mathcal{D}_g$ is the decryption scheme.

This cryptosystem has many nice properties. First, it is *additive homomorphic*. Note we have

- $\mathcal{D}_g(\mathcal{E}_g(m_0, r_0)\mathcal{E}_g(m_1, r_1)) = m_0 + m_1 \bmod n$, and
- $\mathcal{D}_g(\mathcal{E}_g(m_0, r_0)^c) = cm_0 \bmod n$.

Second, it is *semantically secure* under CRA [P99]: assume $m_0, m_1 \in \mathbb{Z}_n$ are two known messages and the ciphertext c is either from m_0 or m_1; note c is from m_0 iff $cg^{-m_0} \bmod n^2$ is an n-th residue. In other words, any successful chosen plaintext attack can be used to decide the composite residuosity, and vice versa.

[1] In [P99], this assumption is named Decisional Composite Residuosity Assumption (DCRA).

3 Cryptographic Schemes

3.1 A basic scheme

W.l.o.g. we assume $N = \ell^2$ for some $\ell \in \mathbb{N}$. Let $x(i,j)_{i,j\in[\ell]}$ denote the bit $x[(i-1)\ell + (j-1) + 1]$, and let $x(i^*, j^*)$ be the bit User wants to learn. Specifically, we treat the database as a 2-hypercube. Also, let $I(t, t_0)$ be an indicating function such that $I(t, t_0) = 1$ iff $t = t_0$, otherwise $I(t, t_0) = 0$.

$\boxed{\text{PIR ON 2-HYPERCUBE}}$

- **Initializing:** User sends $\alpha_t = \mathcal{E}_g(I(t, i^*), r_t)$ and $\beta_t = \mathcal{E}_g(I(t, j^*), s_t)$ to Server for $t \in [\ell]$, where r_t and s_t are chosen uniformly at random from $\mathbb{Z}_n^*$.
- **Filtering:** Server computes $\sigma_i = \prod_{t\in[\ell]} (\beta_t)^{x(i,t)} \bmod n^2$ for $i \in [\ell]$.
- **Splitting-and-then-filtering:** Server splits each σ_i by computing $u_i, v_i \in \mathbb{Z}_n$ such that $\sigma_i = u_i n + v_i$, and then sends $u = \prod_{t\in[\ell]} (\alpha_t)^{u_t} \bmod n^2$ and $v = \prod_{t\in[\ell]} (\alpha_t)^{v_t} \bmod n^2$ to User.
- **Reconstructing:** User computes $x(i^*, j^*) = \mathcal{D}_g(\mathcal{D}_g(u)n + \mathcal{D}_g(v))$.

Lemma 2. *Under CRA,* PIR ON 2-HYPERCUBE *is a one-round implementation of* $\mathsf{1dPIR}$ *with Server-side communication* $2k$ *bits and User-side communication* $2kN^{\frac{1}{2}}$ *bits, where* $k = \lceil 2 \log n \rceil$ *is the security parameter.*

Proof. First, we prove the correctness of the scheme. Note each σ_i is equal to $\mathcal{E}_g(x(i, j^*), \tau_i)$ for some $\tau_i \in \mathbb{Z}_n^*$ since $\mathcal{E}_g$ is additive homomorphic. Similarly, $u = \mathcal{E}_g(u_{i^*}, \tau_u)$ and $v = \mathcal{E}_g(v_{i^*}, \tau_v)$ for some $\tau_u, \tau_v \in \mathbb{Z}_n^*$. Next, note $\mathcal{D}_g(u)n + \mathcal{D}_g(v) = u_{i^*}n + v_{i^*} = \sigma_{i^*} = \mathcal{E}_g(x(i^*, j^*), \tau_{i^*})$ for some $\tau_{i^*} \in \mathbb{Z}_n^*$. Consequently, $\mathcal{D}_g(\mathcal{D}_g(u)n + \mathcal{D}_g(v)) = x(i^*, j^*)$. On the other hand, both the Server-side and the User-side communication complexity can be easily verified.

Next, we prove User's security. Note the only communication sent from User to Server consists of $\{\alpha_t, \beta_t\}_{t\in[\ell]}$. Let $\{\alpha'_t, \beta'_t\}_{t\in[\ell]}$ be the communication induced by another $(i', j') \neq (i^*, j^*)$, $i', j' \in [\ell]$. Clearly, if Server can distinguish these two distributions, it must be the case that Server either can distinguish the distributions of $\{\alpha_t\}_{t\in[\ell]}$ and $\{\alpha'_t\}_{t\in[\ell]}$ or can distinguish the distributions of $\{\beta_t\}_{t\in[\ell]}$ and $\{\beta'_t\}_{t\in[\ell]}$. Suppose Server can distinguish the distributions of $\{\alpha_t\}_{t\in[\ell]}$ and $\{\alpha'_t\}_{t\in[\ell]}$, then by standard hybrid argument we know Server can distinguish the distributions of either $\alpha_{i^*} = \mathcal{E}_g(1, U_{\mathbb{Z}_n^*})$ and $\alpha'_{i^*} = \mathcal{E}_g(0, U_{\mathbb{Z}_n^*})$ or $\alpha_{i'} = \mathcal{E}_g(0, U_{\mathbb{Z}_n^*})$ and $\alpha'_{i'} = \mathcal{E}_g(1, U_{\mathbb{Z}_n^*})$, where $U_{\mathbb{Z}_n^*}$ is the uniform distribution over $\mathbb{Z}_n^*$, as the distributions of α_t and α'_t are identical for $t \in [\ell], t \neq i^*, i'$. Obviously, this implies Server can be used to break the polynomial time indistinguishability of $\mathcal{E}_g$, a contradiction. Since the same argument holds for the case of $\{\beta_t\}_{t\in[\ell]}$ and $\{\beta'_t\}_{t\in[\ell]}$, we finish the proof. ☐

Lemma 3. *Under CRA,* PIR ON 2-HYPERCUBE *is actually an implementation of* $\binom{N}{1}\mathsf{OT}^1$ *against semi-honest Receiver and malicious Sender.*

Proof. Just call Server *Sender* and call User *Receiver*. Note the security of Receiver is guaranteed by the above proof even if Sender is malicious, since the protocol is one-round and starts from Receiver, whose message is independent of Sender's behavior.

Next, note Sender's security is guaranteed if Receiver is semi-honest, as the messages u, v sent from Server to User do not depend on $x(i,j)_{i,j\in[\ell],(i,j)\neq(i^*,j^*)}$. On the other hand, the correctness can be easily verified. ☐

3.2 More than a single bit

Let x' be an array of N entries with each entry containing a $\lfloor \log n \rfloor$-bit string. W.l.o.g., we use $x'[i]_{i\in N}$ to denote the $\lfloor \log n \rfloor$-bit string in the i-th entry of x', and similarly, we use $x'(i,j)$ to denote $x'[(i-1)\ell + (j-1) + 1]$ when N is assumed to be ℓ^2 for some $\ell \in \mathbb{N}$.

Now we make a small modification on our basic scheme: to replace $x(i,t)$ in the second step of the basic scheme by $x'(i,t)$. Clearly, as long as $x'(i^*, j^*) \in \mathbb{Z}_n$, it can be reconstructed in the final step of the modified scheme by the nature of Paillier's cryptosystem. So we have the following.

Corollary 4. *Under CRA,* PIR ON 2-HYPERCUBE *can be modified to implement* $\binom{N}{1}\mathsf{OT}^{\lfloor \log n \rfloor}$ *against semi-honest Receiver and malicious Sender without increasing communication complexity.*

In fact, the above modification directly yields an implementation for $\binom{N}{1}\mathsf{OT}^{\ell}$ for any $\ell > \lfloor \log n \rfloor$. Here the reason is Sender can split each ℓ-bit string into strings of $\lfloor \log n \rfloor$ bits, construct respective arrays, and compute the returning messages separately. Note the protocol is parallelly one-round, and there is no need of additional communication from Receiver since his message can be reused. Moreover, the Sender-side communication is bounded by $2k\lceil \ell / \lfloor \log n \rfloor \rceil = 2\lceil 2 \log n \rceil \lceil \ell / \lfloor \log n \rfloor \rceil$ bits.

Corollary 5. *Under CRA,* PIR ON 2-HYPERCUBE *can be modified to implement* $\binom{N}{1}\mathsf{OT}^{\ell}$ *against semi-honest Receiver and malicious Sender with Sender-side communication* $O(\ell)$ *bits if* $\ell > \lfloor \log n \rfloor$.

3.3 A scheme on c-hypercube

Recall in the basic scheme we treat the database x as a 2-hypercube. Actually, we also can treat the database as a c-hypercube for any integer constant $c > 2$. And by recursive calls, we can achieve communication balance between the Server-side and the User-side, depending on the choice of c.

Here for illustration, let us consider the case $c = 3$, and w.l.o.g. let $N = \ell^3$ for some $\ell \in \mathbb{N}$. Also, we let $x(i,j,\kappa)_{i,j,\kappa\in[\ell]}$ denote the bit $x[(i-1)\ell^2 + (j-1)\ell + (\kappa - 1) + 1]$, and let $x(i^*, j^*, \kappa^*)$ be the bit User wants to learn. Moreover, we keep the definition of $I(t, t_0)$.

PIR ON 3-HYPERCUBE

- **Initializing:** Server and User treat the 3-hypercube database x as ℓ 2-hypercube databases $x(1) = x(i,j,1)_{i,j\in[\ell]}, x(2) = x(i,j,2)_{i,j\in[\ell]}, \cdots, x(\ell) = x(i,j,\ell)_{i,j\in[\ell]}$, while User sends $\gamma_t = \mathcal{E}_g(I(t,\kappa^*), \tau_t)$ to Server for $t \in [\ell]$, where each τ_t is chosen uniformly at random from $\mathbb{Z}_n^*$.
- **Invoking:** User executes PIR ON 2-HYPERCUBE with Server on all $x(d)_{d\in[\ell]}$ in parallel yet omitting the **Reconstructing** step of PIR ON 2-HYPERCUBE and complying the following:
  1. User's message is the same in all executions, with his choice being (i^*, j^*). This says one copy is enough for all executions, and Server should reuse that copy (of $\{\alpha_t, \beta_t\}_{t\in[\ell]}$).
  2. Server does not send to User the pair $(u(d), v(d))$, namely his returning message in PIR ON 2-HYPERCUBE with respect to $x(d)$, after computing it.
- **Splitting-and-then-filtering:** Server instead computes $uu_d, uv_d, vu_d, vv_d \in \mathbb{Z}_n$ such that $u(d) = (uu_d)n + uv_d$ and $v(d) = (vu_d)n + vv_d$, and then sends $uu = \prod_{d\in[\ell]} (\gamma_d)^{uu_d} \bmod n^2$, $uv = \prod_{d\in[\ell]} (\gamma_d)^{uv_d} \bmod n^2$, $vu = \prod_{d\in[\ell]} (\gamma_d)^{vu_d} \bmod n^2$ and $vv = \prod_{d\in[\ell]} (\gamma_d)^{vv_d} \bmod n^2$ to User.
- **Reconstructing:** User computes

$$x(i^*, j^*, \kappa^*) = \mathcal{D}_g(\mathcal{D}_g([\mathcal{D}_g(uu)n + \mathcal{D}_g(uv)])n + \mathcal{D}_g([\mathcal{D}_g(vu)n + \mathcal{D}_g(vv)])).$$

Lemma 6. *Under CRA,* PIR ON 3-HYPERCUBE *is a one-round implementation of* 1dPIR *with Server-side communication* $4k$ *bits and User-side communication* $3kN^{\frac{1}{3}}$ *bits, where* $k = \lceil 2 \log n \rceil$ *is the security parameter.*

Proof. The protocol is one-round as User's sending of $\{\gamma_t\}_{t\in[\ell]}$ can be merged into the executions of PIR ON 2-HYPERCUBE. Next, note that $[\mathcal{D}_g(uu)n + \mathcal{D}_g(uv)] = (uu_{\kappa^*})n + uv_{\kappa^*} = u_{\kappa^*}$ and that $[\mathcal{D}_g(vu)n + \mathcal{D}_g(vv)] = (vu_{\kappa^*})n + vv_{\kappa^*} = v_{\kappa^*}$. So we have

$$\begin{aligned} & \mathcal{D}_g(\mathcal{D}_g([\mathcal{D}_g(uu)n + \mathcal{D}_g(uv)])n + \mathcal{D}_g([\mathcal{D}_g(vu)n + \mathcal{D}_g(vv)])) \\ = \; & \mathcal{D}_g(\mathcal{D}_g(u_{\kappa^*})n + \mathcal{D}_g(v_{\kappa^*})) \\ = \; & x(i^*, j^*, \kappa^*). \end{aligned}$$

On the other hand, the security follows directly the proof for PIR ON 2-HYPERCUBE, while the Server-side communication is straightforward. Finally, the User-side communication follows the fact that User just needs to send one copy of $\{\alpha_t, \beta_t\}_{t\in[\ell]}$, along with $\{\gamma_t\}_{t\in[\ell]}$, to Server. □

In fact, the above scheme itself is a non-black-box reduction from PIR ON 3-HYPERCUBE to PIR ON 2-HYPERCUBE, and the same technique can be applied recursively.

Theorem 7. *Under CRA, We can construct* PIR ON c-HYPERCUBE*, a one-round implementation of* 1dPIR *with Server-side communication* $2^{c-1}k$ *bits and User-side communication* $ckN^{\frac{1}{c}}$ *bits for any integer constant* $c > 3$*, where* $k = \lceil 2 \log n \rceil$ *is the security parameter.*

Proof. (SKETCH ONLY) We just give a high-level description of PIR ON c-HYPERCUBE, which invokes PIR ON $(c-1)$-HYPERCUBE as a sub-routine.

- **Initializing:** Server and User treat the c-hypercube database x as ℓ $(c-1)$-hypercube databases, while User sends the c-th dimensional encrypted indexes to Server.
- **Invoking:** User executes PIR ON $(c-1)$-HYPERCUBE with Server on those ℓ $(c-1)$-hypercube databases in parallel yet omitting the **Reconstructing** step of PIR ON $(c-1)$-HYPERCUBE and complying the following:
  - User's message is the same in all executions and is in accordance with his choice. This says one copy is enough for all executions, and Server should reuse that copy.
  - Server does not send to User his returning messages in PIR ON $(c-1)$-HYPERCUBE after computing them.
- **Splitting-and-then-filtering:** Instead, Server splits the computed returning messages and filters them by multiplying the c-th dimensional encrypted indexes raised to the splits, and then sends the results to User.
- **Reconstructing:** User reconstructs the desired answer by recursive decryptions.

Here the security and the User-side communication can be proved in the same way that we did for PIR ON 3-HYPERCUBE, while the Server-side communication follows the recursive splitting: the communication complexity will increase exponentially to the (constant) dimension of the hypercube by the factor 2. □

Corollary 8. *Under CRA,* PIR ON c-HYPERCUBE *can be modified to implement* $\binom{N}{1}\mathsf{OT}^{\ell}$ *against semi-honest Receiver and malicious Sender for arbitrary* $\ell \in \mathbb{N}$*, while we can use the constant* c *as a parameter to do communication balancing between Sender and Receiver.*

4 Oblivious Transfers against Malicious Players

In the previous section, we show how to implement $\binom{N}{1}\mathsf{OT}^{\lfloor \log n \rfloor}$ against semi-honest Receiver under CRA. In this section, we will transform any such scheme into an implementation of $\binom{N}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ against malicious Receiver. We emphasize that the only zero-knowledge proof setup in our protocol is to prove n is valid, i.e. n is a product of two safe primes of the same length in bits, which is inevitable for any cryptosystem based on the hardness of factoring (e.g. RSA). Besides that, CRA is sufficient to guarantee the security of our construction against malicious players. According to [PS00], we can prove the validity of n in a zero-knowledge manner efficiently with communication complexity and computational complexity being $O(k + \log n)$ and $O(k(k + \log n))$, respectively, where k is the security parameter. Moreover, our protocol can be extended to deal with strings of unrestricted length using the same idea behind Corollary 5.

4.1 $\binom{4}{2}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ against malicious players

We first build a sub-protocol essential to our main construction. Consider computations modulo n^2, where n is the product of two ρ-bit safe primes p and q. Assume Sender has four $(\rho - 1)$-bit strings m_1, m_2, m_3, m_4, and Receiver has two choices $c_1, c_2 \in \{1, 2, 3, 4\}$ and wants to learn m_{c_1} and m_{c_2}. Here is the protocol for them to achieve this task in an oblivious way, with their security being guaranteed even if the other player is malicious. (Note $\rho - 1 = \left\lfloor \frac{\log n}{2} \right\rfloor$.)

$\boxed{\text{2-OUT-OF-4 } (\rho - 1)\text{-BIT STRING OBLIVIOUS TRANSFER}}$

- Receiver uses zero-knowledge proof to convince Sender that his public key n is a product of two safe primes p, q, and computes $a \in \mathbb{Z}_n$ such that $[a + c_1 = 0 \bmod p]$ and $[a + c_2 = 0 \bmod q]$.
- Let $g = n + 1$; Receiver sends $x = \mathcal{E}_g(a, r)$ to Sender, who then verifies $x \in \mathbb{Z}^*_{n^2}$.
- Sender computes the following with computations modulus n^2:

$$y_1 = r_1^n (xg^1)^{\alpha_1} g^{m_1},\ y_2 = r_2^n (xg^2)^{\alpha_2} g^{m_2},\ y_3 = r_3^n (xg^3)^{\alpha_3} g^{m_3},\ y_4 = r_4^n (xg^4)^{\alpha_4} g^{m_4},$$

  where r_1, r_2, r_3, r_4, (resp. $\alpha_1, \alpha_2, \alpha_3, \alpha_4$) are chosen uniformly at random from $\mathbb{Z}^*_n$ (resp. $\mathbb{Z}_n$).
- Sender sends y_1, y_2, y_3, y_4 to Receiver, who then compute

$$m_{c_1} = [\mathcal{D}_g(y_{c_1}) \bmod p],\ m_{c_2} = [\mathcal{D}_g(y_{c_2}) \bmod q].$$

Theorem 9. *Under CRA , for sufficiently large ρ,* 2-OUT-OF-4 $(\rho - 1)$-BIT STRING OBLIVIOUS TRANSFER *is an implementation of $\binom{4}{2}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ against malicious players.*

Proof. First we check Receiver's security. Note we only need to guarantee Receiver's security if he follows the protocol. Clearly, since $x = \mathcal{E}_g(a, r)$ is the only message from Receiver and $\mathcal{E}_g$ is semantically secure, we claim Sender cannot distinguish Receiver's choice. Also because the generation of x does not depend on Sender's behavior (it is the first message in the protocol), we claim Receiver's security holds even if Sender is malicious.

Next, we examine the correctness of the protocol. Similarly, we only need to guarantee the correctness if both players follow the protocol. Note Receiver has the following for $1 \leq i \leq 4$:

$$y_i = [(r^{\alpha_i} r_i)^n g^{\alpha_i(a+i)+m_i} \bmod n^2].$$

In fact, $y_i = [(\Delta_2)^n g^{\Delta_1} \bmod n^2]$, where $\Delta_1 = [\alpha_i(a+i) + m_i \bmod n]$ and $\Delta_2 = [(r^{\alpha_i} r_i) \bmod n]$, since $[g^n = (n+1)^n = 1 \bmod n^2]$ and $[x^n \bmod n^2] = [(x \bmod n)^n \bmod n^2]$ for $x \in \mathbb{N}$. Also, note $\Delta_1 \in \mathbb{Z}_n$ and $\Delta_2 \in \mathbb{Z}_n^*$ (since $r, r_i \in \mathbb{Z}_n^*$). In consequence, we have the following for $1 \le i \le 4$:

$$\mathcal{D}_g(y_i) = \Delta_1 = [\alpha_i(a+i) + m_i \bmod n].$$

So if Receiver follows the protocol, he can certainly get $[m_{c_1} \bmod p]$ and $[m_{c_2} \bmod q]$ since we have $[a + c_1 = 0 \bmod p]$ and $[a + c_2 = 0 \bmod q]$. Moreover, because m_{c_1} (resp. m_{c_2}) is strictly less than p (resp. q), we claim Receiver can obtain the correct values for sure.

Last but most importantly, we have to prove Sender's security against a malicious Receiver, and we will prove that in any case at least two out of $\{y_1, y_2, y_3, y_4\}$ are random in Receiver' view. Recall Receiver has proven to Sender in a zero-knowledge manner that n is valid, i.e. n is a product of two ρ-bit safe primes. Conditioned on such validity of n, we have the following observations.

First, note that given any $c \in \mathbb{Z}_{n^2}^*$ and the factorization of n, one can always compute the corresponding $(a, r) \in (\mathbb{Z}_n, \mathbb{Z}_n^*)$ satisfying $[g^a r^n = c \bmod n^2]$ by the following:

$$a = \mathcal{D}_g(c),\ c_* = cg^{-a},\ r = [c_*^{(n^{-1} \bmod \lambda)} \bmod n].$$

Recall such mapping is bijective (see Section 2). Next, note one can always decide whether a given value is in $\mathbb{Z}_{n^2}^*$ or not (by checking whether it is in $[n^2]$ and is relative prime to n). So we claim

- Receiver cannot send a message $\notin \mathbb{Z}_{n^2}^*$ as Sender can detect it easily, and thus
- Sender can be sure that the only message from Receiver is of the form $[g^a r^n \bmod n^2]$ for some $(a, r) \in (\mathbb{Z}_n, \mathbb{Z}_n^*)$ and that Receiver chooses and knows (a, r) directly or indirectly.

In other words, Receiver's malicious behavior is restricted within the choices of a and r.

Next, the following proof goes for any fixed a, r, and m_1, m_2, m_3, m_4. Note that at least two out of four successive integers are relative prime to n, so we know at least two elements of $\mathcal{A} = \{a_i |\ a_i = a + i \bmod n\}_{1 \le i \le 4}$ are in $\mathbb{Z}_n^*$ and thus have their own inverses. Assume $a_i \in \mathbb{Z}_n^*$ for some $1 \le i \le 4$. We claim y_i is uniformly distributed in Receiver's view, by the following observations:

- $y_i = [(\Delta_2)^n g^{\Delta_1} \bmod n^2]$, where $\Delta_1 = [\alpha_i a_i + m_i \bmod n]$ and $\Delta_2 = [(r^{\alpha_i} r_i) \bmod n]$.
- Δ_1 is uniformly distributed in $\mathbb{Z}_n$. (Since $a_i \in \mathbb{Z}_n^*$ and α_i is uniformly distributed in $\mathbb{Z}_n$, we know $[\alpha_i a_i \bmod n]$ is uniformly distributed in $\mathbb{Z}_n$, and so is Δ_1.)
- When Δ_1 is fixed, Δ_2 is uniformly distributed in $\mathbb{Z}_n^*$. (Since m_i and Δ_1 are fixed, so is $\alpha_i = (\Delta_1 - m_i)(a_i)^{-1}$; since $r \in \mathbb{Z}_n^*$ and r_i is uniformly distributed in $\mathbb{Z}_n^*$, we know Δ_2 is uniformly distributed in $\mathbb{Z}_n^*$.)
- $y_i = [(\Delta_2)^n g^{\Delta_1} \bmod n^2]$ is uniformly distributed in $\mathbb{Z}_{n^2}^*$ due to the bijective mapping.

Consequently, we claim at least two of $\{y_1, y_2, y_3, y_4\}$ are random in Receiver's view, and thus leak no information about the corresponding strings. Since y_1, y_2, y_3, y_4 are the only messages from Sender to Receiver, we finish the proof of Sender's security against a malicious Receiver. $\square$

Lemma 10. 2-OUT-OF-4 $(\rho - 1)$-BIT STRING OBLIVIOUS TRANSFER *can be used to implement* $\binom{2}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ *against malicious players.*

Proof. Assume Sender has two $(\rho-1)$-bit strings x_0, x_1, and Receiver has a choices $b \in \{0,1\}$ and wants to learn x_b.

Let Sender choose two $(\rho-1)$-bit random strings σ_1, σ_2 and execute 2-OUT-OF-4 $(\rho-1)$-BIT STRING OBLIVIOUS TRANSFER with Receiver using the following settings: $m_1 = x_1 \oplus \sigma_1, m_2 = \sigma_1, m_3 = x_2 \oplus \sigma_2, m_4 = \sigma_2, c_1 = 2b+1, c_2 = 2b+2$, where $\oplus$ means bitwise exclusive-or. $\square$

4.2 $\binom{N}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ against malicious players

Under CRA, we now have the following two tools at hand: (Recall $k = \lceil 2\log n \rceil$ is the security parameter and c is a chosen integer constant greater than 1.)

- SCHEME1: A $\binom{N}{1}\mathsf{OT}^{\lfloor \log n \rfloor}$ scheme against semi-honest Receiver and malicious Sender with Sender-side communication $2^{c-1}k$ bits and Receiver-side communication $ckN^{\frac{1}{c}}$ bits.
- SCHEME2: A $\binom{2}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ scheme against malicious Receiver and malicious with Sender-side communication $4k$ bits and Receiver-side communication k bits.

And if we luxuriously treat SCHEME1 as an implementation of $\binom{N}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ against semi-honest Receiver and malicious Sender, we can design a communication-efficient $\binom{N}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ scheme against malicious Receiver and malicious Sender using the technique proposed in [NP99] with the exception that no pseudo-random function is involved in our construction. Details are as follows.

W.l.o.g. we assume $N = 2^t$ for some $t \in \mathbb{N}$, and assume Receiver's choice is $\sigma \in \{0, 1, \cdots, N-1\}$ and Sender's strings are $x_0, x_1, \cdots, x_{N-1} \in \{0,1\}^{\rho-1}$ with $\rho - 1 = \left\lfloor \frac{\log n}{2} \right\rfloor$. Moreover, for $i \in \{0, 1, \cdots, N-1\}$, rewrite i in binary form, i.e. rewrite $i = \sum_{j\in[t]} b_j^i 2^{j-1}$ with $b_j^i \in \{0,1\}$.

$\boxed{\text{1-OUT-OF-}N\ (\rho-1)\text{-BIT STRING OBLIVIOUS TRANSFER}}$

- Sender prepares $2t$ $(\rho-1)$-bit random strings $r_1^0, r_1^1, r_2^0, r_2^1, \cdots, r_t^0, r_t^1$, and computes
$$m_i = x_i \bigoplus_{j\in[t]} r_j^{b_j^i} \text{ for } i \in \{0, 1, \cdots, N-1\}.$$
- Sender and Receiver execute SCHEME2 t times with r_j^0, r_j^1 Sender's inputs and b_j^σ Receiver's choice in the j-th execution.
- Sender and Receiver execute SCHEME1 once with $m_0, m_1, \cdots, m_{N-1}$ Sender's inputs and σ Receiver's choice.
- Receiver computes $x_\sigma = m_\sigma \bigoplus_{j\in[t]} r_j^{b_j^\sigma}$.

Theorem 11. *Under CRA,* 1-OUT-OF-N $(\rho-1)$-BIT STRING OBLIVIOUS TRANSFER *is a one-round implementation of* $\binom{N}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ *secure against malicious players with Sender-side communication* $k(2^{c-1} + 4\log N)$ *bits and Receiver-side communication* $k(cN^{\frac{1}{c}} + \log N)$ *bits.*

Proof. First, the protocol is one-round since the second and the third steps can be executed in parallel. Next, the correctness can be easily verified, and the communication overheads come from those t executions of SCHEME2, which only consist of $4k\log N$ bits from Sender and $k\log N$ bits from Receiver (as $t = \log N$).

The security of Receiver (against malicious Sender) follows the properties of SCHEME1 and SCHEME2. On the other hand, to prove Sender's security (against malicious Receiver), it is enough to consider the case that Sender sends all $m_0, m_1, \cdots, m_{N-1}$ to Receiver directly in the third step, and the proof is standard since SCHEME2 can guarantee security against malicious players: in this case $N-1$ elements out of $\{m_0, m_1, \cdots, m_{N-1}\}$ must be random in Receiver's view, hence nothing about the corresponding $N-1$ strings is leaked. □

Theorem 12. *Under CRA,* 1-OUT-OF-N $(\rho-1)$-BIT STRING OBLIVIOUS TRANSFER *can be modified to implement* $\binom{N}{1}\mathsf{OT}^{\ell}$ *against malicious Receiver and malicious Sender for large* ℓ *with Sender-side communication* $O(\ell \log N)$ *bits.*

Proof. It is enough to mention the following observation: we can extend SCHEME2 to deal with strings of unrestricted length in the way we did for SCHEME1 in Corollary 5. Note such extension is secure even if Receiver is malicious, since it reuses Receiver's message instead of asking more from Receiver. In fact, the reused message can be thought as a commitment from Receiver. By applying both extensions (of SCHEME1 and SCHEME2), we obtain the desired one-round implementation. □

4.3 Discussions

In [AIR01] (and independently in [NP01]), a method to construct oblivious transfer protocols against malicious players using any additive homomorphic encryption scheme was proposed, with a constraint that the mathematical structure underlies the additive homomorphic property must be a *field*. And we believe there is no trivial solution to get over such a constraint.

Clearly, the additive homomorphic property of Pailliar's cryptosystem is over $\mathbb{Z}_n$, which is not a field. So we applied certain tricks, on top of the ideas from [AIR01] (and [NP01]), to deal with malicious players in Section 4.1.

On the other hand, it was mentioned in Section 4.2 that we applied a technique similar to that proposed in [NP99] to construct $\binom{N}{1}\mathsf{OT}^{\lfloor \frac{\log n}{2} \rfloor}$ against malicious players, and we want to elaborate this point in detail.

In fact, [NP99] showed a way to construct a communication-efficient $\binom{N}{1}\mathsf{OT}^{\ell}$ protocol using the combination of the following: $\mathsf{1dPIR}$, $\binom{2}{1}\mathsf{OT}^{k}$, and a pseudo-random function, where k is the security parameter and $\ell \in \mathbb{N}$ is unrestricted. Roughly speaking, the N strings $x_0, x_1, \cdots, x_{N-1}$ in our scheme were interpreted as inputs to a pseudo-random function in [NP99], and the real messages are exclusive-or-ed with the pseudo-random outputs. In this way, no restriction was put on ℓ. (And $\mathsf{1dPIR}$ takes the role of SCHEME1, which in our case is also an implementation of $\mathsf{1dPIR}$.)

However, by the nature of our schemes, there is no such need of a pseudo-random function, and we especially want our schemes to be *self-contained* within CRA. So we abandon the usage of a pseudo-random function, which, however, is a good alternative in practice [IKNP03].

Finally, we mention there are oblivious transfer protocols against malicious players even without any zero-knowledge setup [AIR01, NP01], whose security are based on the hardness of decisional Diffie-Hellman assumption. However, our scheme has the advantage of having communication efficiency under a single computational assumption, which is the main consideration of this paper.

Final remark. Recently, two groups of researchers (Y. Ishai, E. Kushilevitz, R. Ostrovsky and M. Freedman, K. Nissim, B. Pinkas) independently discovered a similar approach to build efficient PIR protocols using Paillier's cryptosystem, and the communication complexity of their schemes is the same with ours, yet their results haven't been published. We are informed by the first group.

References

[A01] D. Asonov, "Private information retrieval: an overview and current trends," Manuscript, 2001. (Available online at http://www.dbis.informatik.hu-berlin.de/∼asonov/.)

[AIR01] W. Aiello, Y. Ishai, and O. Reingold, "Priced oblivious transfer: how to sell digital goods," *Eurocrypt 2001*, pp. 119–135.

[BM84] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits," *SIAM Journal on Computing, 13(4)*: pp. 850–864, 1984.

[CGG02] D. Catalano, R. Gennaro, and N. H.-Graham, "Paillier's trapdoor function hides up to O(n) bits," *Journal of Cryptology, 15(4)*: pp. 251–269, 2002.

[CGGN01] D. Catalano, R. Gennaro, N. H.-Graham, and P. Nguyen, "Paillier's cryptosystem revisited," *ACM Conference on Computer and Communications Security 2001*, pp. 206–214.

[CMO00] G. Crescenzo, T. Malkin, and R. Ostrovsky, "Single database private information retrieval implies oblivious transfer," *Eurocrypt 2000*, pp. 122–138.

[CMS99] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," *Eurocrypt'99*, pp. 402–414.

[CNS02] D. Catalano, P. Nguyen, and J. Stern, "The hardness of hensel lifting: the case of RSA and discrete logarithm," *Asiacrypt 2002*, pp. 299–310.

[CS02] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," *Eurocrypt 2002*, pp. 45–64.

[DJ01] I. Damgard and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," *Public Key Cryptography 2001*, pp. 119–136.

[G98] O. Goldreich, "Secure multi-party computation," Manuscript, 1998. (Available online at http://www.wisdom.weizmann.ac.il/∼oded/.)

[G00] S. Galbraith, "Elliptic curve Paillier schemes," *Journal of Cryptology, 15(2)*: pp. 129–138, 2000.

[GM84] S. Goldwasser and S. Micali, "Probabilistic encryption," *JCSS, 28(2)*: pp. 270–299, 1984.

[GMMV03] D. Galindo, S. Mollevi, P. Morillo, and J. Villar, "A practical public key cryptosystem from Paillier and Rabin schemes," *Public Key Cryptography 2003*, pp. 279–291.

[IKNP03] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank, "Extending oblivious transfers efficiently," *CRYPTO 2003*.

[K88] J. Kilian, "Founding cryptography on oblivious transfer," *STOC'88*, pp. 20–31.

[KO97] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: single database, computationally-private information retrieval," *FOCS'97*, pp. 364–373.

[KO00] E. Kushilevitz and R. Ostrovsky, "One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval," *Eurocrypt 2000*, pp. 104–121.

[KT03] K. Kurosawa and T. Takagi, "Some RSA-based encryption schemes with tight security reduction," *Asiacrypt 2003*.

[NP99] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," *STOC'99*, pp. 245–254.

[NP01] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols," *SODA 2001*, pp. 448–457.

[P99] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *Eurocrypt'99*, pp. 223–238.

[PP99] P. Paillier and D. Pointcheval, "Efficient public-key cryptosystems provably secure against active adversaries," *Asiacrypt 1999*, pp. 165–179.

[PS00] G. Poupard and J. Stern, "Short proofs of knowledge for factoring," *Public Key Cryptography 2000*, pp. 147–166.

[R81] M. Rabin, "How to exchange secrets by oblivious transfer," Technical Report TR-81, Harvard University, 1981.

[ST02] K. Sakurai and T. Takagi. "New semantically secure public-key cryptosystems from the RSA-primitive," *Public Key Cryptography 2002*, pp. 1–16.