# An Oblivious Transfer Protocol with Log-Squared Communication

Helger Lipmaa

[1] Cybernetica AS, Lai 6, 51005 Tartu, Estonia
[2] Institute of Computer Science, University of Tartu, J. Liivi 2, 50409 Tartu, Estonia

**Abstract.** We propose a one-round 1-out-of-$n$ computationally-private information retrieval protocol for $\ell$-bit strings with low-degree polylogarithmic receiver-computation, linear sender-computation and communication $\Theta(k \cdot \log^2 n + \ell \cdot \log n)$, where $k$ is a possibly non-constant security parameter. The new protocol is receiver-private if the underlying length-flexible additively homomorphic public-key cryptosystem is IND-CPA secure. It can be transformed to a one-round computationally receiver-private and information-theoretically sender-private 1-out-of-$n$ oblivious-transfer protocol for $\ell$-bit strings, that has the same asymptotic communication and is private in the standard complexity-theoretic model.
**Keywords.** Computationally-private information retrieval, length-flexible additively homomorphic public-key cryptosystem, oblivious transfer.

## 1 Introduction

During a 1-out-of-$n$ computationally-private information retrieval protocol for $\ell$-bit strings, $\mathrm{CPIR}_\ell^n$, Receiver retrieves an entry from Sender's database $S = (S[1], \ldots, S[n])$, $S[j] \in \{0,1\}^\ell$, so that a computationally bounded Sender will not obtain any information on which element was retrieved. The first and up to now the only $\mathrm{CPIR}_\ell^n$ protocol, $\mathsf{CMS}_\ell^n$, with polylogarithmic in $n$ communication was proposed in [CMS99]. Alternatively, based on an earlier work by Kushilevitz and Ostrovsky [KO97], Julien P. Stern [Ste98] proposed another family—that we call $\mathsf{HomCPIR}_\ell^n(\alpha)$—of $\mathrm{CPIR}_\ell^n$ protocols, based on an arbitrary IND-CPA secure additively homomorphic public-key cryptosystem. If say $n < 2^{40}$, then Stern's protocol is quite communication-efficient. In particular, for all realistic values of $n$ and $\ell$, it is vastly more communication-efficient than $\mathsf{CMS}_\ell^n$.

However, the communication of $\mathsf{HomCPIR}_\ell^n(\alpha)$ is not polylogarithmic, and may be even more importantly, it has superpolylogarithmic Receiver's computation and superlinear Sender's computation in $n$. In particular, Sender's superlinear computation makes Stern's protocol inapplicable for say $n > 2^{15}$. This can be compared with essentially constant-time Receiver's computation and linear-time Sender's computation in the linear-communication $\mathrm{CPIR}_\ell^n$ protocols of [NP01,AIR01]. Construction of an efficient-in-practice (this involves both communication-efficiency and computation-efficiency) and yet polylogarithmic $\mathrm{CPIR}_\ell^n$ protocol has been a major open problem.

In this paper, we propose a new $\mathrm{CPIR}_\ell^n$ protocol with log-squared communication that has a very low computational overhead. It takes advantage of the concept of length-flexible additively homomorphic (LFAH) public-key cryptosystems [DJ01,DJ03]. Recall that a LFAH public-key cryptosystem has an additional length parameter $s \in \mathbb{Z}^+$,

such that given a public and private key pair of the receiver and a random value belonging to an $s$-independent set, the encryption algorithm maps $sk$-bit plaintexts, for any $s$ and for a security parameter $k$, to $(s+\xi)k$-bit ciphertexts for some small integer $\xi \geq 1$; $\xi = 1$ in the case of the cryptosystem from [DJ01]. This can be compared to the conventional additively homomorphic public-key cryptosystems [Pai99] that map $k$-bit plaintexts to $\eta k$-bit ciphertexts for some $\eta \geq 2$.

Now, assume that $s = \lceil \ell/k \rceil$. Assume the existence of an LFAH public-key cryptosystem with the mentioned properties. We show that for any $\alpha \in [\log n]$, there exists a $\mathrm{CPIR}_\ell^n$ protocol $\mathsf{LFCPIR}_\ell^n(\alpha)$ with communication $(\alpha \cdot (s + \frac{\xi}{2}(\alpha+1))(n^{1/\alpha}-1) + s + \alpha\xi) \cdot k$ bits. In particular, in the asymptotically optimal case $\alpha = \log n$, the communication of $\mathsf{LFCPIR}_\ell^n(\log n)$ is $(\frac{\xi}{2} \cdot \log^2 n + (s + \frac{3\xi}{2}) \cdot \log n + s) \cdot k = \Theta(k \cdot \log^2 n + \ell \cdot \log n)$ bits. Moreover, if $\ell \geq k \cdot \log n$, then $\mathsf{LFCPIR}_\ell^n(\log n)$ has communication $\Theta(\ell \cdot \log n)$ bits with the constant in the $\Theta$-expression being arbitrary close to 1; this is very close to the communication of non-private information retrieval, $\lceil \log n \rceil + \ell$. An important property of our protocols is that they are simple to understand and to implement.

Additionally, we describe some variants of our basic protocol that are especially efficient for particular values of $\ell$ and $n$, and that enable to balance communication and computation. For example, we describe an $\mathrm{CPIR}_\ell^n$ protocol with communication $(1+\xi)((n-1)k + \ell)$; this results in close-to-optimal communication in the case of small databases but long documents.

If one uses a fast exponentiation algorithm, Sender's work in a slight variant of $\mathsf{LFCPIR}_\ell^n(\log n)$ is equivalent to $\Theta(n\ell) \cdot k^{2+o(1)}$ bit-operations; this is optimal in $n$ up to a multiplicative constant. Receiver's work is low-degree polylogarithmic in $n$, $\Theta((k \cdot \log n + \ell)^{2+o(1)})$ bit-operations, and therefore also close to optimal.

Our results indicate that in the case of $\mathrm{CPIR}_\ell^n$ protocols, one should not overemphasise complexity-theoretic notions like polylogarithmicity, but instead study the communication of a protocol in a very concrete framework. This is best illustrated by the fact that for $n \leq 2^{40}$, the only previous polylogarithmic $\mathrm{CPIR}_\ell^n$ protocol by Cachin, Micali and Stadler requires more communication then just transferring the whole database. On the other hand, we do not deny that having polylogarithmic communication is important in theoretic frameworks. The new protocols, proposed in this paper, are both polylogarithmic ("good in theory") and require less communication than any of the previous $\mathrm{CPIR}_\ell^n$ protocols for practically any values of $n$ and $\ell$ ("good in practice").

All previous protocols that use LFAH public-key cryptosystems utilise encryptions only under a single, although possible very large, value of the length parameter $s$. A transcript of the $\mathsf{LFCPIR}_\ell^n(\alpha)$ protocol includes encryptions of interrelated plaintexts under different values of the length parameter. This use of LFAH public-key cryptosystems is novel and therefore interesting by itself. We define a new security requirement for cryptosystems, $\alpha$-IND-LFCPA-security, and show that known IND-CPA secure LFAH public-key cryptosystems are secure in the sense of $\alpha$-IND-LFCPA (under a tight reduction), and that $\mathsf{LFCPIR}_\ell^n(\alpha)$ is secure under a tight reduction to the $\alpha$-IND-LFCPA-security of the underlying public-key cryptosystem, or under a looser reduction to the IND-CPA-security of the underlying public-key cryptosystem.

We briefly discuss the potentially stronger setting where one needs security against adversaries that work in time $\mathrm{poly}(n\ell)$. Since the Decisional Composite Residuosity Problem modulo $M$ can be solved in time $\exp(O(1)\log^{1/3} M \cdot (\log\log M)^{2/3})$ by using general number field sieve, one must have $k = \Omega(\log^{3-o(1)}(n\ell))$. Thus, if security against such adversaries is required, $\mathsf{LFCPIR}^n_\ell(\log n)$ has communication $\Omega(\log^{3-o(1)}(n\ell) \cdot \log^2 n + \ell \cdot \log n)$. If one comes up with a suitable cryptosystem that has better security guarantees, then the exponent $3-o(1)$ can be improved to $2-o(1)$ or even to 1. Additionally, we show that $\mathsf{LFCPIR}^n_\ell(\log n)$, if based on the cryptosystems from [DJ01,DJ03], has communication $\Theta(\kappa^{3-o(1)} \cdot \log^2 n + \ell \cdot \log n)$, where $\kappa$ is a security parameter that corresponds to the *exponential* security level.

Finally, we show that one can transform $\mathsf{LFCPIR}^n_\ell(\alpha)$ to a computationally receiver-private and information-theoretically sender-private one-round $\mathrm{OT}^n_\ell$ protocol, with log-squared communication, that is secure in the standard complexity-theoretic model.

An early version of this paper (that in particular had the description of $\mathsf{LFCPIR}^n_\ell(\alpha)$) was posted on the IACR eprint server [Lip04] in Spring 2004. The conference version has been shortened due to the lack of space. The full version is available from [Lip04].

## 2 Preliminaries

For a $t \in \mathbb{Z}^+$, let $[t]$ denote the set $\{1, \ldots, t\}$. All logarithms in this paper will be on base 2, unless explicitly mentioned. Let $e$ be the base of the natural logarithm, that is, $\ln e = 1$. For a distribution (random variable) $X$, let $x \leftarrow X$ denote the assignment of $x$ according to $X$. We often identify sets with the uniform distributions on them, and algorithms with their output distributions, assuming that the algorithm that outputs this distribution is clear from the context or just straightforward to construct. Let $k$ and $\kappa$ be two security parameters, where $k$ corresponds to the superpolynomial security (breaking some primitive is hard in time $\mathrm{poly}(k)$) and $\kappa$ corresponds to the exponential security (breaking some primitive is hard in time $2^{o(\kappa)}$). Denote $L_M[a,b] := \exp(a(\ln M)^b \cdot (\ln\ln M)^{1-b})$. Throughout this paper, we denote Sender's database size by $n$, assume that database elements belong to $\{0,1\}^\ell = \mathbb{Z}_{2^\ell}$ for some fixed positive integer $\ell$, and denote $s := \lceil \ell/k \rceil$. We denote $\mathsf{sqrtlog}(a,b) := \sqrt{\log_a b}$.

Assume that $M = p_1 p_2$ is a product of two large primes. A number $z$ is said to be an $M$-th residue modulo $M^2$ if there exists a number $y \in \mathbb{Z}_{M^2}$ such that $z = y^M \mod M^2$. The *decisional composite residuosity problem* [Pai99] (DCRP) is to distinguish $M$-th residues from $M$-th non-residues. The fastest known way to break DCRP is to factor the modulus $M$, which can be done in time $O(L_M[(64/9)^{1/3} + o(1), 1/3])$ by using general number field sieve.

*A length-flexible additively homomorphic (LFAH) public-key cryptosystem* is a tuple $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where (a) $\mathsf{Gen}$ is a key generation algorithm, that on input $1^k$, returns $(\mathsf{sk}, \mathsf{pk})$, where $\mathsf{sk}$ is a secret key and $\mathsf{pk}$ is a public key, (b) $\mathsf{Enc}$ is an encryption algorithm, that on input $(\mathsf{pk}, s, m, r)$, where $\mathsf{pk}$ is a public key, $s \in \mathbb{Z}^+$ is a length parameter, $m$ is a plaintext and $r$ is a random coin, returns a ciphertext $\mathsf{Enc}^s_{\mathsf{pk}}(m; r)$, and (c) $\mathsf{Dec}$ is an decryption algorithm that on input $(\mathsf{sk}, s, c)$, where $\mathsf{sk}$ is a secret key, $s$ is a length parameter and $c$ is a ciphertext, returns a plaintext $\mathsf{Dec}^s_{\mathsf{sk}}(c)$. For any $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^k)$ and for any $s \in \mathbb{Z}^+$, $\mathsf{Enc}^s_{\mathsf{pk}} : \mathcal{M}_s \times \mathcal{R} \to \mathcal{C}_s$ and $\mathsf{Dec}^s_{\mathsf{pk}} : \mathcal{C}_s \to$

$\mathcal{M}_s$, where $\mathcal{C}_s$ is the ciphertext space and $\mathcal{M}_s$ is the plaintext space corresponding to $s$, and $\mathcal{R}$ is the $s$-independent randomness space. We require that for some positive integer $a$, $\mathcal{C}_s \subseteq \mathcal{M}_{s+a}$ for every $s$; we assume that $\xi$ is the minimal among such $a$'s. Length-flexible cryptosystems not satisfying the latter requirement exist but are not interesting in the context of our application. An LFAH public-key cryptosystem $\Pi$ is *additively homomorphic* if for any key pair $(\mathsf{sk}, \mathsf{pk})$, any length parameter $s$, any $m, m' \in \mathcal{M}_s = \mathbb{Z}_{\sharp\mathcal{M}_s}$ and any $r, r' \in \mathcal{R}$, $\mathsf{Enc}^s_{\mathsf{pk}}(m; r) \cdot \mathsf{Enc}^s_{\mathsf{pk}}(m'; r') = \mathsf{Enc}^s_{\mathsf{pk}}(m + m'; r \circ r')$, where $\cdot$ is a multiplicative group operation in $\mathcal{C}_s$, $+$ is addition in $\mathbb{Z}_{\sharp\mathcal{M}_s}$, and $\circ$ is a groupoid operation in $\mathcal{R}$. We assume that $k = \log \sharp\mathcal{M}_1$ is the security parameter. For the sake of simplicity, in our computations we will assume that $\sharp\mathcal{M}_s = (\sharp\mathcal{M}_1)^s$ with $\log \sharp\mathcal{M}_s = sk$, and that $\sharp\mathcal{C}_s = \sharp\mathcal{M}_{s+\xi}$.

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a LFAH public-key cryptosystem. We define the advantage of a randomised algorithm $A$ in breaking its IND-CPA security as follows: $\mathsf{Adv}^{\mathsf{indcpa}}_{\Pi,k}(A) := 2 \cdot \left|\Pr[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^k), (m_0, m_1, s) \leftarrow A(\mathsf{pk}), b \leftarrow \{0,1\}, r \leftarrow \mathcal{R} : A(\mathsf{pk}, m_0, m_1, s, \mathsf{Enc}^s_{\mathsf{pk}}(m_b; r)) = b] - \frac{1}{2}\right|$. Here, the probability is taken over the random coin tosses of $\mathsf{Gen}$, $A$, $\mathsf{Enc}^s_{\mathsf{pk}}$ and over the choice of $b$ and $r$. We say that $\Pi$ is $(\varepsilon, \tau)$-*secure in the sense of IND-CPA* if $\mathsf{Adv}^{\mathsf{indcpa}}_{\Pi,k}(A) \leq \varepsilon$ for any randomised algorithm $A$ that works in time $\tau$. If $\tau(k)$ is polynomial in $k$ and $\varepsilon(k)$ is negligible in $k$, then we sometimes just say that $\Pi$ is secure in the sense of IND-CPA.

The Damgård-Jurik cryptosystem DJ01 from PKC 2001 [DJ01] was the first published IND-CPA secure LFAH public-key cryptosystem. Assume that $M = p_1 p_2$ is an RSA modulus. Here, for a fixed length parameter $s$, $\mathcal{M}_s = \mathbb{Z}_{M^s}$, $\mathcal{R} = \mathbb{Z}^*_M$ and $\mathcal{C}_s = \mathbb{Z}^*_{M^{s+1}}$, thus $\log \sharp\mathcal{C}_s / \log \sharp\mathcal{M}_s \approx 1 + 1/s$ and $\xi = 1$. Encryption is defined by $\mathsf{Enc}^s_{\mathsf{pk}}(m; r) := (1 + M)^m \cdot r^{M^s} \mod M^{s+1}$, where $r \leftarrow \mathbb{Z}_M$. The DJ01 cryptosystem is additively homomorphic since $\mathsf{Enc}^s_{\mathsf{pk}}(m_1; r_1) \cdot \mathsf{Enc}^s_{\mathsf{pk}}(m_2; r_2) = \mathsf{Enc}^s_{\mathsf{pk}}(m_1 + m_2; r_1 r_2)$. The DJ01 LFAH public-key cryptosystem is secure in the sense of IND-CPA, assuming that the DCRP is hard [DJ01]. The Damgård-Jurik cryptosystem DJ03 from ACISP 2003 [DJ03] is slightly less efficient than DJ01, with $\log \sharp\mathcal{C}_s / \log \sharp\mathcal{M}_s \approx 1 + 2/s$, that is, with $\xi = 2$.

IND-CPA secure LFAH public-key cryptosystems have been used before, in particular, to implement multi-candidate electronic voting [DJ01,DJ03] and large-scale electronic auctions [LAN02] over large plaintext spaces. We use LFAH cryptosystems in a more complicated setup that requires the transfer of encryptions of related plaintexts modulo different length parameters during the same protocol instance.

During a (single-server) 1-*out-of-*$n$ *computationally-private information retrieval* ($\mathrm{CPIR}^n_\ell$) protocol for $\ell$-bit strings, Receiver fetches $S[q]$ from the database $S = (S[1], \ldots, S[n])$, $S[j] \in \{0,1\}^\ell$, so that a computationally bounded Sender does not know which entry Receiver is learning. We do not require Sender to commit to or even "know" a database to which Client's search is effectively applied. Such a relaxation is standard in the case of protocols like oblivious transfer, computationally-private information retrieval and oblivious keyword search; our security definitions correspond closely to the formalisation given in [NP01,AIR01].

Formally, a one-round $\mathrm{CPIR}^n_\ell$ protocol $\Gamma$ is a triple of algorithms, $(\mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$, corresponding to the two messages of the protocol and the recovery phase. Query and Transfer are randomised and Recover is, in the

context of this paper, deterministic. Let $\mathcal{R}_Q$ and $\mathcal{R}_T$ be two distributions, associated with $\Gamma$, and let $k$ be the security parameter. As usually, we assume that the database size $n$ is known to Receiver. The first message, $\mathsf{msgq} \leftarrow \mathsf{Query}(1^k, q, n; r_Q)$, of a protocol run is by Receiver Rec, where $q$ is his input (index to the database), $n$ is the database size and $r_Q \leftarrow \mathcal{R}_Q$ is a new random value. The second message is by Sen, who replies with $\mathsf{msgt} \leftarrow \mathsf{Transfer}(1^k, S, \mathsf{msgq}; r_T)$, where $S$ is her input (the database), $\mathsf{msgq}$ is the first message of the protocol, and $r_T \leftarrow \mathcal{R}_T$ is a new random value. After the second message, Receiver returns his private output $\mathsf{Recover}(1^k, q, \mathsf{msgq}, \mathsf{msgt})$. In general, the *communication* of $\Gamma$ is equal to $|\mathsf{msgq}| + |\mathsf{msgt}|$. However, we make a convention that transferring Receiver's public key—that is a part of several well-known $\mathrm{CPIR}_\ell^n$ protocols—does not increase the communication of $\Gamma$. We can do this because the usually very short public key can often be transferred before the actual data itself becomes available; the key can also be shared between many protocol runs. However, we will not prove security in this setting. Note that the communication complexity of information retrieval, without any privacy requirements and with no additional information on the structure of the data that would enable to compress it, is $\lceil \log n \rceil + \ell$.

We say that a $\mathrm{CPIR}_\ell^n$ protocol $\Gamma = (\mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$ is *correct* if for any $n$, $S \in \{0,1\}^{n\ell}$, $q \in [n]$, $\mathsf{Recover}(1^k, q, \mathsf{msgq}, \mathsf{msgt}) = S[q]$, given that $\mathsf{msgq} \leftarrow \mathsf{Query}(1^k, q, n; r_Q)$ for some $r_Q \in \mathcal{R}_Q$ and $\mathsf{msgt} \leftarrow \mathsf{Transfer}(1^k, S, \mathsf{msgq}; r_T)$ for some $r_T \in \mathcal{R}_T$. For a randomised algorithm $A$ executing Sender's part in a $\mathrm{CPIR}_\ell^n$ protocol $\Gamma$ and for a positive integer $n$, define

$$\mathsf{Adv}^{\mathsf{cpir}}_{\Gamma,n,k}(A) := 2 \cdot \max_{q_0,q_1} \left| \Pr \left[ \begin{array}{l} b \leftarrow \{0,1\}, r_Q \leftarrow \mathcal{R}_Q : \\ A(1^k, q_0, q_1, n, \mathsf{Query}(1^k, q_b, n; r_Q)) = b \end{array} \right] - \frac{1}{2} \right|$$

to be the scaled advantage over random coin-tossing that $A$ has in guessing, which of the two possible choices $q_0$ and $q_1$ was used by the receiver, after observing a single query from Receiver. Here, $q_0$ and $q_1$ are supposed to be valid inputs to $\mathsf{Query}(\cdot, \cdot, n; \cdot)$. The probability is taken over the coin tosses of $A$ and Query, and over the choices of $b$ and $r_Q$. We call $\Gamma$ a $(\tau, \varepsilon)$-*receiver-private* $\mathrm{CPIR}_\ell^n$ *protocol*, if $\mathsf{Adv}^{\mathsf{cpir}}_{\mathsf{Rec},n,k}(A) \leq \varepsilon(k, n, \ell)$ for any probabilistic algorithm $A$ that works in time $\tau(k, n, \ell)$. In Sect. 4, we study an alternative definition where $\tau$ is an unspecified value with $\tau > \mathrm{poly}(n\ell)$.

The first $\mathrm{CPIR}_1^n$ protocol with sublinear communication, $O(2^{\mathsf{sqrtlog}(2,n)\cdot\mathsf{sqrtlog}(2,k)})$, was proposed by Kushilevitz and Ostrovsky in [KO97]. The first $\mathrm{CPIR}_1^n$ protocol $\mathsf{CMS}_1^n$ with polylogarithmic communication was proposed by Cachin, Micali and Stadler in [CMS99]. The security of the $\mathsf{CMS}_1^n$ protocol is based on the $\Phi$ Assumption that basically states that there exists a constant $f$, such that given a large composite $M$ with unknown factorisation and a small prime $p$ with $M \approx p^f$, it is hard to decide whether $p \mid \phi(M)$. The $\mathsf{CMS}_1^n$ protocol has receiver-side communication $2\kappa^f + \kappa^4$ (Receiver sends a triple $(m, x, Y)$ with $\log m = \log x = \kappa^f$ and $\log Y = \kappa^4$) and sender-side communication $\kappa^f$ (Sender sends a value $r$ with $\log r = \kappa^f$). Its total communication is $\kappa^4 + 3\kappa^f = \Omega(\log^8 n + \log^{2f} n)$ for some constant $f$ and a security parameter $\kappa > \log^2 n$. In particular, its communication depends on $f$, existence of which is conjectured by the $\Phi$ Assumption. No hypothesis about the value of $f$ was made in [CMS99], except that $f \geq 4$ to provide security against Coppersmith's algorithm that efficiently factors $m$ on inputs $(p, m)$, where $p > m^{1/4}$ is a prime such that $p \mid \phi(m)$.

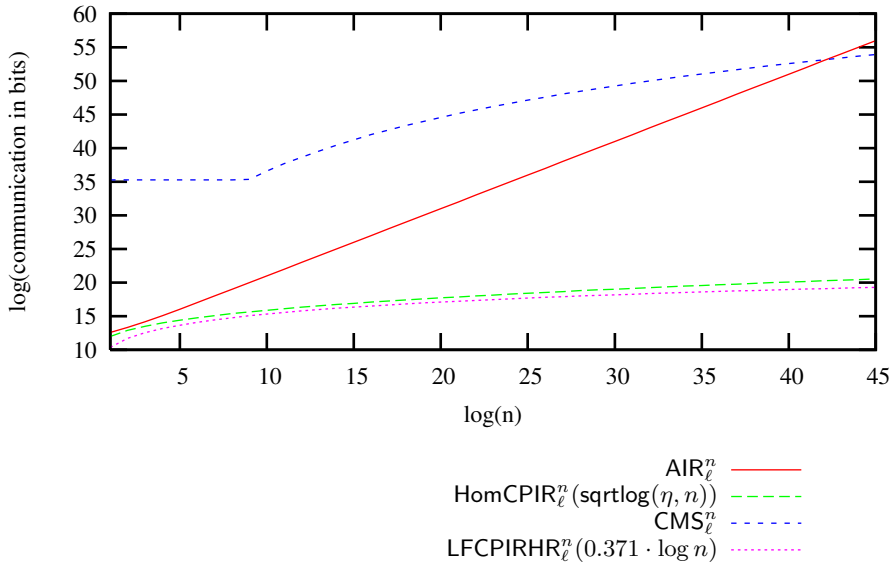


**Fig. 1.** Logarithm of communication of some of the previously known CPIR's on the logarithmic scale in $n$, assuming that $k = 1024$ and $\eta = 2$. (Except for the $\mathsf{CMS}^n_\ell$ protocol that has a security parameter $\kappa = \max(80, \log^2 n)$.) Here, $\ell = 1024$

One can transform $\mathsf{CMS}^n_1$ to a $\mathrm{CPIR}^n_\ell$ protocol by running it $\ell$ times in parallel (with the same Receiver's query); thus $\mathsf{CMS}^n_\ell$ has communication $\Omega(\ell \cdot \log^{2f} n + \log^8 n)$. Even if polylogarithmic, the communication of the $\mathsf{CMS}^n_\ell$ protocol is larger than just sending the database to Receiver for all relevant database sizes. (See Fig. 1.) In the $\mathsf{CMS}^n_\ell$ protocol, Receiver's computation is polylogarithmic in $n$.

The Kushilevitz-Ostrovsky $\mathrm{CPIR}^n_\ell$ was generalised by Julien P. Stern [Ste98]; Stern's protocol was later rediscovered by Chang [Cha04]. Stern's $\mathrm{CPIR}^n_\ell$ is based on an arbitrary IND-CPA secure additively homomorphic cryptosystem $\Pi$. Similarly to our previous convention, $\mathcal{M}$ is $\Pi$'s plaintext space and $\mathcal{C}$ is $\Pi$'s ciphertext space. Let $\eta := \lceil \log \sharp\mathcal{C} / \log \sharp\mathcal{M} \rceil$ be the *ciphertext expansion ratio* of $\Pi$; $\eta = 2$ for the Paillier cryptosystem [Pai99] and for the Damgård-Jurik cryptosystem from PKC 2001 [DJ01] and $\eta \in \{2, 3\}$ for another cryptosystem by Damgård and Jurik [DJ03]. W.l.o.g., assume that Sender's database $S = (S[1], \ldots, S[n])$ contains $n = \lambda^\alpha$ entries from $\{0,1\}^\ell$ for some positive integer $\lambda$ and for $\alpha \in [\log_\eta n]$. As always, let $s := \lceil \ell / k \rceil$. As shown in [Ste98], there exists an $\mathrm{CPIR}^n_\ell$ protocol $\mathsf{HomCPIR}^n_\ell(\alpha)$ with the communication $(\eta\alpha n^{1/\alpha} + s\eta^\alpha) \cdot k$ bits. In particular, for $\delta := \mathsf{sqrtlog}(\eta, n)$, $\mathsf{HomCPIR}^n_\ell(\delta)$ has communication $(\eta\delta + s)\eta^\delta \cdot k$ bits. ([Ste98,Cha04] erroneously claims that the communication of $\mathsf{HomCPIR}^n_\ell(\delta)$ is $\Theta(\eta^\delta) \cdot k$.) While even in the optimal case, $\mathsf{HomCPIR}^n_\ell(\alpha)$ has superpolylogarithmic communication, $\mathsf{HomCPIR}^n_\ell(\delta)$ is significantly more communication-efficient than $\mathsf{CMS}^n_\ell$ for all relevant database sizes $n \leq 2^{80}$. (See Fig. 1.) Finally, Sender's (resp., Receiver's) computation is dominated

by $\Theta(sn2^\delta)$ (resp., $\Theta(sn\delta 2^\delta)$) $k$-bit exponentiations. This means that Stern's CPIR is computationally less efficient than the Cachin-Micali-Stadler CPIR.

A $\mathrm{CPIR}_\ell^n$ protocol $(\mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$ is an *(computationally receiver-private and information-theoretically sender-private)* $1$*-out-of-*$n$ *oblivious transfer protocol for* $\ell$*-bit strings* (an $\mathrm{OT}_\ell^n$ protocol) if also Sender's privacy is guaranteed. For the formal definition, we make a comparison to the ideal implementation, using a trusted third party that receives $S$ from Sender, receives $q$ from Receiver, and sends $S[q]$ to Receiver. We assume that Receiver receives garbage (that is, a random value from some $S$-independent set $T$) if $q \notin [n]$. We do not need an explicit security definition of a secure oblivious transfer protocol in this paper. (See, for example, [NP01].)

## 3 New $\mathrm{CPIR}_\ell^n$ with Log-Squared Communication

In this section, we use a LFAH public-key cryptosystem $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ to improve over the concrete and the asymptotic communication (and computation) of $\mathsf{HomCPIR}_\ell^n(\alpha)$, by presenting a family $\mathsf{LFCPIR}_\ell^n(\alpha)$ of $\mathrm{CPIR}_\ell^n$ protocols. As always, we define $s := \lceil \ell/k \rceil$.

The basic idea of Protocol 1 is relatively simple. Fix $\alpha \in [\log n]$. Assume that the database $S = (S[1], \ldots, S[n])$ is arranged as an $\alpha$-dimensional $\lambda_1 \times \cdots \times \lambda_\alpha$ hyperrectangle, for some positive integers $\lambda_j$ that will be defined later. W.l.o.g., we assume that $n = \prod_{j=1}^{\alpha} \lambda_j$. In the simplest case, $\alpha = \log n$ and $\lambda_j = 2$, then the database is just arranged on a $2 \times \cdots \times 2$ hypercube. We index every element $S[i]$ in the database by its coordinates $(i_1, \ldots, i_\alpha)$ on this hyperrectangle, where $i_j \in \mathbb{Z}_{\lambda_j}$. I.e.,

$$S(i_1, \ldots, i_\alpha) := S[i_1 \cdot \prod_{j=2}^{\alpha} \lambda_j + i_2 \cdot \prod_{j=3}^{\alpha} \lambda_j + \cdots + i_{\alpha-1} \cdot \lambda_\alpha + i_\alpha + 1]$$

for $i_j \in \mathbb{Z}_{\lambda_j}$. Analogously, Receiver's query is $q = (q_1, \ldots, q_\alpha)$ with $q_j \in \mathbb{Z}_{\lambda_j}$.

We use homomorphic properties of $\Pi$ to create a new database $S_1$ that has $\alpha - 1$ dimensions, such that $S_1(i_2, \ldots, i_\alpha)$ is equal to an encryption of $S_0(q_1, i_2, \ldots, i_\alpha)$, where $S_0 = S$. We use this procedure repeatedly for $j \in [\alpha]$, to create $(\alpha - j)$-dimensional databases $S_j$, where the $(s + j\xi)k$-bit element $S_j(i_j, \ldots, i_\alpha)$ encrypts $j$ times the value $S(q_1, \ldots, q_{j-1}, i_j, \ldots, i_\alpha)$. At the end of the $\alpha$th iteration, Sender has a single $(s+\alpha\xi)k$-bit element $S_\alpha$ that is an $\alpha$-times encryption of $S(q_1, \ldots, q_\alpha) = S[q]$. Therefore, it suffices for Sender to just transfer one value $S_\alpha$, with length $|S_\alpha| = (s + \alpha\xi)k$, to Receiver. After that, Receiver recovers $S[q]$ by decrypting $S_\alpha$ $\alpha$ times. Thus, the basic idea of the new protocol is similar to that of $\mathsf{HomCPIR}_\ell^n(\alpha)$. Since $\Pi$ is length-flexible, instead of dividing every intermediate ciphertext into $\eta$ chunks as in the case of $\mathsf{HomCPIR}_\ell^n(\alpha)$, we additively increase the length of the plaintexts. Our underlying observation is that $\mathsf{Enc}_{\mathsf{pk}}^{s+\xi}(m_2; r_2)^{\mathsf{Enc}_{\mathsf{pk}}^{s}(m_1; r_1)} = \mathsf{Enc}_{\mathsf{pk}}^{s+\xi}(m_2 \cdot \mathsf{Enc}_{\mathsf{pk}}^{s}(m_1; r_1); r_3) \in \mathcal{M}_{s+2\xi}$ for any $m_1 \in \mathcal{M}_s$, $m_2 \in \mathcal{M}_{s+\xi}$ and $r_1, r_2 \in \mathcal{R}$, and for some $r_3 \in \mathcal{R}$. In particular, it is equal to an encryption of zero if $m_2 = 0$ and to a double-encryption of $m_1$ if $m_2 = 1$. Protocol 1 depicts the new $\mathsf{LFCPIR}_\ell^n(\alpha)$ protocol with parameters, optimised for large values of $\ell$. Note that here $\mathcal{R}_Q = \mathcal{R}^{\sum_{j \in [\alpha]} \lambda_j}$ and $\mathcal{R}_T = \emptyset$.

*Private Input*: Receiver has $n$ and $q = (q_1, \ldots, q_\alpha)$, Sender has $S$.
*Private Output*: Receiver obtains $S(q_1, \ldots, q_\alpha)$.

**Receiver,** $\mathsf{Query}(1^k, q, n; \mathcal{R}_Q)$**:**
Generate a key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^k)$.
For $j \leftarrow 1$ to $\alpha$ do, for $t \leftarrow 0$ to $\lambda_j - 1$ do:
Generate $r_{jt} \leftarrow \mathcal{R}$.
If $q_j = t$ then set $b_{jt} \leftarrow 1$ else set $b_{jt} \leftarrow 0$.
Set $\beta_{jt} \leftarrow \mathsf{Enc}_{\mathsf{pk}}^{s+(j-1)\xi}(b_{jt}; r_{jt})$.
Send $(\mathsf{pk}, (\beta_{jt})_{j\in[\alpha], t\in\mathbb{Z}_{\lambda_j}})$ to Sender.

**Sender,** $\mathsf{Transfer}(1^k, S_0, \mathsf{msgq}; \mathcal{R}_T)$**:**
For $j \leftarrow 1$ to $\alpha$ do
For $i_{j+1} \leftarrow 0$ to $\lambda_{j+1} - 1, \ldots, i_\alpha \leftarrow 0$ to $\lambda_\alpha - 1$ do:
Set $S_j(i_{j+1}, \ldots, i_\alpha) \leftarrow \prod_{t\in\mathbb{Z}_{\lambda_j}} \beta_{jt}^{S_{j-1}(t, i_{j+1}, \ldots, i_\alpha)}$.
Send $S_\alpha$ to Receiver.

**Receiver** $\mathsf{Recover}(1^k, q, \mathsf{msgq}, S'_\alpha)$**:**
For $j \leftarrow \alpha$ downto 1 do: Set $S'_{j-1} \leftarrow \mathsf{Dec}_{\mathsf{sk}}^{s+(j-1)\xi}(S'_j)$.
Output $S'_0$.

**Protocol 1:** Protocol $\mathsf{LFCPIR}_\ell^n(\alpha)$ (non-optimised version), for fixed $\Pi$ and fixed $s$. Here, $\beta_{jt}, S_j(i_{j+1}, \ldots, i_\alpha) \in \mathcal{C}_{s+(j-1)\xi}$

We make the next simple observation that Sender can compute $\beta_{j,\lambda_j-1}$ by himself, by setting $\beta_{j,\lambda_j-1} \leftarrow \mathsf{Enc}_{\mathsf{pk}}^{s+(j-1)\xi}(1;0)/\prod_{t=0}^{\lambda_j-2} \beta_{jt}$; this optimisation is valid since $\prod_{t=1}^{\lambda_j-1} \beta_{jt}$ is always an encryption of 1. Therefore, in Protocol 1, Receiver does not have to send $\beta_{j,\lambda_j-1}$ to Sender. In the most practical case, where $\lambda_j = 2$, this optimisation reduces communication by a factor of 2. In this case, this optimisation also substantially simplifies some of the oblivious transfer protocols, mentioned later in Sect. 4. In the following, when we talk about the $\mathsf{LFCPIR}_\ell^n(\alpha)$ protocol, we always assume that one applies this optimisation. Moreover, recall that the communication of a $\mathrm{CPIR}_\ell^n$ protocol does not include $\mathsf{pk}$.

**Theorem 1.** *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an LFAH public-key cryptosystem. Assume that $\mathcal{M}_{s+1} < 2^\ell \leq \mathcal{M}_s$ for some fixed $s \geq 1$, that Receiver has private input $q$ and Sender has private input $S = (S[1], \ldots, S[n])$. Assume that $\lambda_j = n^{1/\alpha}$ for all $j \in [\alpha]$.*

1. *For every $\alpha \in [\log n]$, there exists a correct $\mathrm{CPIR}_\ell^n$ protocol $\mathsf{LFCPIR}_\ell^n(\alpha)$ with the receiver-side and the sender-side communication $\alpha(s + (\alpha+1)\frac{\xi}{2})(n^{1/\alpha} - 1) \cdot k$ and $(\alpha\xi + s) \cdot k$ bits.*
2. *$\mathsf{LFCPIR}_\ell^n(\log n)$ has receiver-side communication $(\frac{\xi}{2} \cdot \log^2 n + (s + \frac{\xi}{2}) \cdot \log n) \cdot k = \Theta(k \cdot \log^2 n + \ell \cdot \log n)$ and sender-side communication $(\xi \cdot \log n + s) \cdot k = \Theta(k \cdot \log n + \ell)$. In this case, Receiver's workload is $\tau_{\mathsf{Rec}} = \Theta((s^{2+o(1)} \cdot \log n + \xi s \cdot \log^{2+o(1)} n + \xi^{2+o(1)} \cdot \log^{3+o(1)} n)k^{2+o(1)}$ and Sender's workload is $\tau_{\mathsf{Sen}} := \Theta(n) \cdot (sk)^{2+o(1)}$.*

*Proof. Correctness:* clear, since $S_j(i_{j+1},\ldots,i_\alpha)$ is an $j$-times encryption of $S(q_1,\ldots,q_j,i_{j+1},\ldots,i_\alpha)$ and thus $S'_{\alpha-1}=S_{\alpha-1}(q_\alpha)$, $S'_{\alpha-2}=S_{\alpha-2}(q_{\alpha-1},q_\alpha)$, …, $S'_{i-1}=S_{i-1}(q_i,\ldots,q_\alpha)$, …, and $S'_0=S(q_1,\ldots,q_\alpha)$.

*Communication:* The receiver-side communication $|\mathsf{msgq}|$ is

$$\sum_{j=1}^{\alpha}\sum_{t=1}^{\lambda_j-1}(s+j\xi)k=\sum_{j=1}^{\alpha}(s+j\xi)\cdot(n^{1/\alpha}-1)\cdot k=\alpha\cdot(s+(\alpha+1)\xi/2)(n^{1/\alpha}-1)\cdot k$$

bits. This is asymptotically optimal in $s\cdot\log n$ if $\alpha=\log n$.

*Computation (in the case (2)):* Sender's work is dominated by $2^{\log n-j}$ exponentiations modulo $M^{s+j\xi}$ for every $j\in[2,\alpha]$. Assume that a $k$-bit exponentiation can be done in time $\Theta(k^a)$ for some $a$. Then, Sender's workload is dominated by $n\cdot\sum_{j=2}^{\log n}2^{-j}\cdot\Theta((s+j\xi)^a k^a)$ bit-operations. Asymptotically in $n$, this is equal to $\Theta(n)\cdot(sk)^a$; fast exponentiation algorithms result in Sender's time $\Theta(n)\cdot(sk)^{2+o(1)}$. Receiver must do $\lambda_j-1$ encryptions $\mathsf{Enc}_{\mathsf{pk}}^{s+(j-1)\xi}$ for any $j\in[n]$. Thus, Receiver's work is $\sum_{j=1}^{\log n}\Theta((s+(j-1)\xi)^a k^a)=\sum_{j=1}^{\log n}\Theta((s^a+(j\xi)^a)k^a)=\Theta((s^{2+o(1)}\log n+\xi s\log^{2+o(1)}n+\xi^{2+o(1)}\cdot\log^{3+o(1)}n)k^{2+o(1)}$ bit-operations, if using asymptotically fast exponentiation algorithms. □

It is surprising that such a seemingly simple modification of $\mathsf{HomCPIR}_\ell^n(\alpha)$ results in the important asymptotic improvement, stated by Thm. 1: namely, using an LFAH public-key cryptosystem where $(s+j\xi)k$-bit plaintexts are encrypted to $(s+(j+1)\xi)k$-bit ciphertexts, we achieve communication $\Theta(k\cdot\log^2 n+\ell\cdot\log n)$, while using an additively homomorphic public-key cryptosystem where $(s+j)k$-bit plaintexts are encrypted to $\eta(s+j)k$-bit ciphertexts, enabled [Ste98] to get communication $\Theta(\ell\cdot\mathsf{sqrtlog}(\eta,n)\cdot 2^{\mathsf{sqrtlog}(\eta,n)}+k\cdot\mathsf{sqrtlog}(\eta,n)\cdot 2^{\mathsf{sqrtlog}(\eta,n)})$. Additionally, $\mathsf{LFCPIR}_\ell^n(n)$ is also more computation-efficient. These substantial improvements are possible because a LFAH public-key cryptosystem is essentially a new primitive and not just another off-the-shelf homomorphic cryptosystem.

We will prove the receiver-privacy of this protocol later in Section 4. In the rest of this section, we propose some quite important optimisations.

*Optimisation for long documents and in Sender's computation.* For long documents, $\mathsf{LFCPIR}_\ell^n(\alpha)$ gains even more on the competitors than for short documents. For $\ell=\Omega(k\cdot\log n)$, the asymptotic communication of $\mathsf{LFCPIR}_\ell^n(\alpha)$ is $\Theta(\ell\cdot\log n)$ that is asymptotically optimal. Note that the constant inside the $\Theta$ expression gets arbitrary close to 1. If $\ell>k$, then one can execute $s=\lceil\ell/k\rceil$ instances of $\mathsf{LFCPIR}_{2^k}^n(\alpha)$'s in parallel, with the same Receiver's message, with the receiver-side and the sender-side communication of respectively $\sum_{j=1}^{\alpha}\sum_{t=1}^{\lambda_j-1}(1+j\xi)k=\sum_{j=1}^{\alpha}(1+j\xi)\cdot(n^{1/\alpha}-1)\cdot k=\alpha\cdot(1+(\alpha+1)\xi/2)(n^{1/\alpha}-1)\cdot k$ and $s(\alpha\xi+1)\cdot k$ bits. We call this version $\mathsf{LFCPIRBIG}_\ell^n(\alpha)$. For $\alpha=\log n$ it has $(s-1)(\xi-1)k\cdot\log n$ bits more computation than $\mathsf{LFCPIRBIG}_\ell^n(\alpha)$, however, Sender's computation is only $\Theta(n\ell)\cdot k^{2+o(1)}$, which is an important gain compared to $\mathsf{LFCPIR}_\ell^n(\log n)$. If needed, one can optimise asymptotic communication of $\mathsf{LFCPIRBIG}_\ell^n(\alpha)$ in $\ell$ by setting $\alpha\leftarrow 1$, then the communication is $(1+\xi)(n-1+s)\cdot k=\Theta(n\cdot k+\ell)$ bits; however, $\mathsf{LFCPIRBIG}_\ell^n(1)$

is the same as $\mathsf{HomCPIR}^n_\ell(1)$. A variant like $\mathsf{LFCPIRBIG}^n_\ell(\mathsf{sqrtlog}(2,n))$ seems to perform reasonably well in the practice, with typically less communication than $\mathsf{HomCPIR}^n_\ell(\mathsf{sqrtlog}(2,n))$.

*Optimisation for short documents.* For short documents, we can apply a different optimisation strategy. As always, let $s := \lceil \ell/k \rceil$. Let $W$ be Lambert's $W$ function, that is, $W$ satisfies the functional identity $W(x)e^{W(x)} = x$. First, we can use $\mathsf{LFCPIR}^n_\ell(\alpha_0 \cdot \log n)$ with $\alpha_0 = \ln 2/(W(-2e^{-2}) + 2) \approx 0.435$; this results in the minimal communication $\approx (0.371 \cdot \xi \cdot \log^2 n + 1.706 \cdot s \cdot \log n + 1.288 \cdot \xi \cdot \log n + s) \cdot k$ for small values of the length parameter $s$. Second, we can redefine the values of $\lambda_j$ as $\lambda_j \leftarrow ((s+\alpha)!/s!)^{1/\alpha} \cdot (s+j)^{-1} \cdot n^{1/\alpha}$. This choice of $\lambda_j$ results in the minimal value of $\sum_{j=1}^{\alpha}(\lambda_j - 1)(s+j) = \sum_{j=1}^{\alpha} \lambda_j(s+j) - \alpha(s + (\alpha+1)/2)$ under the constraint that $\prod_{j=1}^{\alpha} \lambda_j = n$. (In practice, we must round $\lambda_i$-s to the nearest integers. For the simplicity of exposition, we will not explicitly mention such issues anymore.) Call the resulting instantiation of the protocol $\mathsf{LFCPIRHR}^n_\ell(\alpha)$. $\mathsf{LFCPIRHR}^n_\ell(\alpha)$ has receiver-side and sender-side communication of respectively $((s+\alpha)!/s!)^{1/\alpha} \cdot \alpha \cdot (n^{1/\alpha} - 1) \cdot k$ and $(s+\alpha) \cdot k$ bits. In particular, $\mathsf{LFCPIRHR}^n_\ell(\alpha_0 \cdot \log n)$ has communication $\approx (0.273 \cdot \log n + (0.627 \cdot s + 0.314) \cdot \log\log n + O(1))k \cdot \log n = \Theta(k \cdot \log^2 n + \ell \cdot \log n \cdot \log\log n)$. For $s = 1$, $\mathsf{LFCPIRHR}^n_\ell(\alpha)$ is asymptotically approximately $1.348$ times more communication-efficient than $\mathsf{LFCPIR}^n_\ell(\alpha)$.

If $z := \lfloor sk/\ell \rfloor > 1$, then one can use the next optimisation. Execute $\mathsf{LFCPIR}^n_\ell(\bar{\alpha})$ with the query $\bar{q} := \lfloor q/z \rfloor$ and the database $\bar{S} = (\bar{S}[1], \ldots, \bar{S}[\lfloor n/z \rfloor])$, where $\bar{S}[j]$ is the concatenation of $z$ different consequent elements $S[\lceil j/z \rceil], \ldots, S[\lceil j/z \rceil + z - 1]$ from the database $S$. Fixing $\bar{\alpha} = \log(n/z)$, one can construct a $\mathrm{CPIR}^n_\ell$ with total communication $\approx (0.273 \cdot \log^2(n\ell/(sk)) + 0.435 \cdot s \cdot \log(n\ell/(sk)) \cdot \log\log(n\ell/(sk)) + O(1)) \cdot k$. This optimisation can be quite important in practice. In the extreme case when $n = k = 1024$ and $\ell = 1$, the optimised version is 100 times more communication-efficient than the unoptimised version.

## 4 On Security of LFCPIR And Transformation to OT

In all $\mathrm{CPIR}^n_\ell$ protocols, proposed in Sect. 3, we have the next novel adversarial situation. Given a LFAH public-key cryptosystem $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, the adversary obtains encryptions of interrelated plaintexts by using potentially different values of the length parameter $s$, where $s$ is possibly chosen by herself. It must be the case that the adversary obtains no new knowledge about the encrypted values. Clearly, security in this adversarial situation is a generally desirable feature of LFAH public-key cryptosystems whenever it might be the case that the adversary obtains different-length encryptions of related plaintexts. This may happen almost always, except when all participants are explicitly prohibited to encrypt related messages by using different values of $s$. Therefore, we will introduce the corresponding security requirement formally and prove that some of the previously introduced LFAH public-key cryptosystems have *tight* security also in such an adversarial situation.

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a LFAH public-key cryptosystem. We define the advantage of a randomised algorithm $A$ in breaking $\Pi$'s $\alpha$-IND-LFCPA security as fol-

lows:

$$\mathsf{Adv}^{\mathsf{lf\text{-}indcpa}}_{\Pi,k}(A,\alpha) := 2 \cdot \left| \Pr \left[ \begin{array}{l} (\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^k), \\ (m_0, m_1, s_1, \ldots, s_\alpha) \leftarrow A(\mathsf{pk}), b \leftarrow \{0,1\}, \\ c_1 \leftarrow \mathsf{Enc}^{s_1}_{\mathsf{pk}}(m_b \mod \sharp\mathcal{M}_{s_1}; \mathcal{R}), \ldots, \\ c_\alpha \leftarrow \mathsf{Enc}^{s_\alpha}_{\mathsf{pk}}(m_b \mod \sharp\mathcal{M}_{s_\alpha}; \mathcal{R}) : \\ A(\mathsf{pk}, m_0, m_1, s_1, \ldots, s_\alpha, c_1, \ldots, c_\alpha) = b \end{array} \right] - \frac{1}{2} \right| .$$

(To prove the security of $\mathsf{LFCPIR}^n_\ell(\alpha)$, we could use a slightly weaker assumption where $s_1, \ldots, s_\alpha$ are not chosen by $A$; it is sufficient to consider the case $s_j = s + (j-1)\xi$. We omit discussion because of the lack of space.) Here, probability is taken over random coin tosses of $\mathsf{Gen}$, $\mathsf{Enc}^{s_j}_{\mathsf{pk}}$, $A$ and over the choice of $b$ and of random elements from $\mathcal{R}$. We say that $\Pi$ is $(\varepsilon,\tau)$-*secure in the sense of* $\alpha$-*IND-LFCPA* if $\mathsf{Adv}^{\mathsf{lf\text{-}indcpa}}_{\Pi,k}(A,\alpha) \le \varepsilon$ for any probabilistic algorithm $A$ that works in time $\tau$. If $\tau(k)$ is polynomial in $k$ and $\varepsilon(k)$ is negligible in $k$, then we just say that $\Pi$ is secure in the sense of $\alpha$-IND-LFCPA. We omit $\alpha$ if $\alpha$ may be any polynomial in $k$.

By a standard hybrid argument, $(\alpha\varepsilon, \tau - O(\alpha))$-security in the sense of $\alpha$-IND-LFCPA follows from the $(\varepsilon,\tau)$-security in the sense of IND-CPA. However, since IND-LFCPA security is such a basic notion for LFAH public-key cryptosystems, it makes sense to prove the IND-LFCPA security directly, without the intermediate $\alpha$-times security degradation. Next, we will show that for both well-known LFAH public-key cryptosystems (DJ01 and DJ03), IND-LFCPA security follows from IND-CPA security under a tight reduction. First, we prove the following lemma that is motivated by the observation that IND-LFCPA is a potentially stronger security notion than IND-CPA only in situations where the adversary cannot herself compute, given $\mathsf{Enc}^s_{\mathsf{pk}}(m;\mathcal{R})$, encryptions of related plaintexts with different values of the length parameter $s$.

**Lemma 1.** *Assume* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a LFAH cryptosystem that is* $(\varepsilon,\tau)$*-secure in the sense of IND-CPA. Assume there exists an algorithm* $\mathsf{Shorten}$*, such that for all* $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^k)$*, any* $s_1 < s_2$*, any* $m \in \mathcal{M}_{s_1}$ *and any* $r \in \mathcal{R}$*,* $\mathsf{Shorten}(\mathsf{pk}, s_1, s_2, \mathsf{Enc}^{s_2}_{\mathsf{pk}}(m;r)) = \mathsf{Enc}^{s_1}_{\mathsf{pk}}(m;\mathcal{R})$*. Assume* $\mathsf{Shorten}$ *can be computed in time* $t_{\mathsf{Shorten}}(k,s_2)$*. Then* $\Pi$ *is* $(\varepsilon, \tau - \alpha \cdot t_{\mathsf{Shorten}}(k, s_{\max}) - O(\alpha))$*-secure in the sense of* $\alpha$*-IND-LFCPA where* $s_{\max}$ *is the largest* $s_i$ *that an admissible adversary can choose.*

*Proof.* Really, assume $A$ is an adversary who breaks the $\alpha$-IND-LFCPA security in time $\tau'$ and with probability $\varepsilon$. Construct the next adversary $M^A$ that breaks the IND-CPA security of $\Pi$: Obtain a new random public key $\mathsf{pk}$, send this to $A$. $M$ asks $A$ to produce $(m_0, m_1, s_1, \ldots, s_\alpha)$. Assume that $s_1 \le s_2 \le \cdots \le s_\alpha \le s_{\max}$. Give $(m_0, m_1, s_\alpha)$ to the black box, who returns $c_\alpha \leftarrow \mathsf{Enc}^{s_\alpha}_{\mathsf{pk}}(m_b;\mathcal{R})$. Compute $c_i \leftarrow \mathsf{Shorten}(\mathsf{pk}, s_i, s_\alpha, \mathsf{Enc}^{s_2}_{\mathsf{pk}}(m_b;\mathcal{R}))$ for $i \in [\alpha - 1]$. Send $(c_1, \ldots, c_\alpha)$ to $A$, obtain her guess $b'$. Return $b'$. Clearly, if $A$ has guessed correctly then $b' = b$. □

For both DJ01 and DJ03 it is straightforward to construct the required function $\mathsf{Shorten}$. In the case of the DJ01, $\mathsf{Enc}^{s_1}_{\mathsf{pk}}(m;\mathcal{R}) = (\mathsf{Enc}^{s_2}_{\mathsf{pk}}(m;r) \mod M^s) \cdot \mathsf{Enc}^{s_1}_{\mathsf{pk}}(0;\mathcal{R})$. In the case of the DJ03 cryptosystem, $\mathsf{Enc}^s_{\mathsf{pk}}(m;r) = (g^r \mod M, (1+M)^m (h^r$

$\bmod M)^{M^s} \bmod M^{s+1}$). Therefore, given $\mathsf{Enc}_{\mathsf{pk}}^{s_2}(m;r) = (a,b)$, one can compute $\mathsf{Enc}_{\mathsf{pk}}^{s_1}(m;\mathcal{R}) = (a, b \bmod M^{s_1}) \cdot \mathsf{Enc}_{\mathsf{pk}}^{s_1}(0;\mathcal{R})$. We would get a similar security result, if there existed an efficient function $\mathsf{Expand}$, such that for $s_2 < s_1$, and for any $m \in \mathcal{M}_{s_2}$, $\mathsf{Expand}(\mathsf{pk}, s_1, s_2, \mathsf{Enc}_{\mathsf{pk}}^{s_2}(m;\mathcal{R})) = \mathsf{Enc}_{\mathsf{pk}}^{s_1}(m;\mathcal{R})$. As we show in the full version, the existence of such a function would additionally result in a $\mathrm{CPIR}_\ell^n$ protocol with logarithmic communication. Now, we can prove the next result.

**Theorem 2.** *Fix $n$ and $\alpha \in [\log n]$. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a LFAH public-key cryptosystem that is $(\varepsilon,\tau)$-secure in the sense of $\alpha$-IND-LFCPA, where $\tau \gg \tau_{\mathsf{Sen}}$. Fix $s$. Then $\mathsf{LFCPIR}_\ell^n(\alpha)$ is $(\varepsilon,\tau')$-receiver-private. Here, $\tau' = \tau - \tau_{\mathsf{Rec}} - O(\alpha \cdot (sk)^{1+o(1)})$, where $\tau_{\mathsf{Rec}}$ is the time to execute the honest Receiver.*

*Proof.* Assume that some adversary $A$ that works in time $\tau$ breaks the receiver-privacy of $\mathsf{LFCPIR}_\ell^n(\alpha)$ with probability $\varepsilon$. More precisely, $A$ generates a key pair $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Gen}(1^k)$. Given $\mathsf{pk}$ and an arbitrary $(q_0,q_1)$, $A$ generates $S$ and sends $n$ to Receiver. Receiver picks a random bit $\widehat{b}$ and sends the first message $\mathsf{Query}(1^k, q_{\widehat{b}}, n; r_Q) = (\mathsf{pk}, (\beta_{jt})_{jt})$ of the $\mathsf{LFCPIR}_\ell^n(\alpha)$ protocol, where $r_Q$ is randomly chosen from $\mathcal{R}_Q$, to $A$. $A$ outputs a guess $\widehat{b}'$, such that $2 \cdot |\Pr[\widehat{b} = \widehat{b}'] - \frac{1}{2}| \geq \varepsilon$. Next, we construct a machine $M$ that uses $A$ as an oracle to break the $\alpha$-IND-LFCPA security of $\Pi$ with probability $\mathsf{Adv}_{\Pi,k}^{\mathsf{indcpa}}(M^A) = \varepsilon$. That is, given a random key pair $(\mathsf{sk},\mathsf{pk})$, $M$ comes up with a message pair $(m_0,m_1)$ and length parameters $(s_1,\ldots,s_\alpha)$, such that after seeing $\mathsf{Enc}_{\mathsf{pk}}^{s_i}(m_b;\mathcal{R})$ for a random $b \leftarrow \{0,1\}$ and for $i \in [\alpha]$, $M$ outputs a bit $b'$, such that $2 \cdot |\Pr[b = b'] - \frac{1}{2}| \geq \varepsilon$.

$M$ does the next: Let Receiver to generate $(\mathsf{pk},\mathsf{sk})$, obtain $\mathsf{pk}$ and forward it to $A$. Obtain $(q_0,q_1)$ where $q_i = (q_{i1},\ldots,q_{i\alpha})$. Assume that $q_0$ and $q_1$ differ in the coordinate set $\mathcal{J}$. $M$ sets $m_0 \leftarrow 0$, $m_1 \leftarrow 1$ and asks for a challenge on $(m_0,m_1,(s+(j-1)\xi)_{j\in\mathcal{J}})$. For a random $b \leftarrow \{0,1\}$, $M$ obtains the challenge tuple $(c_j \leftarrow \mathsf{Enc}_{\mathsf{pk}}^{s+(j-1)\xi}(m_b;\mathcal{R}))_{j\in\mathcal{J}}$. $M$ constructs the query $(\beta_{jt})_{j,t}$ exactly as in Protocol 1, except that when $j \in \mathcal{J}$, he sets $\beta_{j,q_{0j}} \leftarrow c_j$ and $\beta_{j,q_{1j}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}^{s+(j-1)\xi}(1;0)\cdot c_j^{-1}$. Therefore, $(\mathsf{pk},(\beta_{jt})_{j,t}) = \mathsf{Query}(1^k, q_b, n; \mathcal{R}_Q)$. $M$ sends $(\mathsf{pk},(\beta_{jt})_{j,t})$ to $A$ and obtains her guess $\widehat{b}'$. $M$ returns $b' = \widehat{b}'$. Clearly, $b = b'$ if $A$ guessed correctly. Therefore, $M$ has success probability $\varepsilon$. $M$'s time is equal to $\tau + \tau_{\mathsf{Rec}} + O(\alpha \cdot (sk)^{1+o(1)})$. □

This result means in particular that $\mathsf{LFCPIR}_\ell^n(\alpha)$ is receiver-private (a) under loose reduction with $\alpha$-times security degradation, in the case $\Pi$ is an arbitrary IND-CPA secure LFAH public-key cryptosystem; (b) under tight reduction to the underlying cryptographic problems, in the case $\Pi$ is DJ01 or DJ03.

**On Concrete Versus Polynomial Security.** It is necessary to use concrete security (that is, always talking about adversaries, working in time $\tau$ and breaking a primitive with probability $\varepsilon$) when one wants to be able to precisely quantify the value of the used security parameter. However, recall that the input size of Sender in a $\mathrm{CPIR}_\ell^n$ protocol is $n\ell$ and that Sender's computation is at least linear in $n\ell$ (this follows directly from the privacy requirement). Clearly, an adversary should be given time that is vastly larger than the time, given to the honest Sender. In Thm. 2, we resolved this

by requiring that $\tau \gg \tau_{\mathsf{Sen}}$. Alternatively, one can require that no adversary is able to break $\mathrm{CPIR}_\ell^n$ in time, polynomial in $n\ell$, with a non-negligible probability in $n\ell$. Assume also that the underlying hard problem, with inputs $M$ of size $k$, can be broken in time $L_M[a,b]$. In the case of $\mathsf{LFCPIR}_\ell^n(\alpha)$, when based on the DJ01 or the DJ03 cryptosystem, $b = 1/3$. Then, it is necessary that $L_M[a,b] = \omega((n\ell)c)$ for every constant $c$, or that $k^b \log^{1-b} k = \omega(\log(n\ell))$. Omitting the logarithmic factor, we get that $k = \Omega(\log^{1/b}(n\ell))$. Therefore, if we want security against adversaries, working in time $\mathrm{poly}(n\ell)$, when basing $\mathsf{LFCPIR}_\ell^n(\alpha)$ on the DCRP, we must assume that $k = \Omega(\log^{3-o(1)}(n\ell))$ and thus the communication of the $\mathsf{LFCPIR}_\ell^n(\log n)$ becomes $\Theta(\log^{3-o(1)}(n\ell) \cdot \log^2 n + \ell \cdot \log n)$. While such an analysis is usually not necessary in stand-alone applications of computationally-private information retrieval, there are theoretical settings where polynomial security is desired (e.g., when a CPIR protocol is a subprotocol of a higher level application).

Alternatively, one can define another security parameter, $\kappa$, corresponding to the desideratum that breaking the $\mathrm{CPIR}_\ell^n$ protocol should be hard in time $2^{o(\kappa)}$, and then expressing the communication in the terms of $\kappa$. Based on the hypothesis that the best attack against the DCRP is the general number field sieve, it means that $k \cdot (\ln k)^2 = \Omega(\frac{9(\ln 2)^2\kappa^3}{64}) = \Omega(\kappa^3)$ and thus $\mathsf{LFCPIR}_\ell^n(\alpha)$, based on any LFAH public-key cryptosystem that relies on the DCRP being hard, has communication $\Theta(\kappa^{3-o(1)} \cdot \log^2 n + \ell \cdot \log n)$. In particular, this captures reasonably well the natural requirement that the adversary should be able to spend at least as much time as Sender: in practice, given large enough $\kappa$ (say, $\kappa = 80$), we may assume that a honest Sender always spends considerably less time than $2^\kappa$ units. This also means that $n$ is restricted to be considerably smaller than $2^\kappa$, but we do not see now problems with that in practice; it is hard to imagine anybody executing a $\mathrm{CPIR}_\ell^n$ protocol with $n$ larger than $2^{40}$! Additionally, this gives us another argument why small sender-side computation is important for a $\mathrm{CPIR}_\ell^n$ protocol. As mentioned before, $\mathsf{LFCPIR}_\ell^n(\cdot)$ does better than $\mathsf{HomCPIR}_\ell^n(\cdot)$ also in this sense.

**Oblivious Transfer with Log-Squared Communication.** We can use one of several existing techniques to transform the $\mathsf{LFCPIR}_\ell^n(\alpha)$ protocol into an oblivious transfer protocol. For these techniques to apply, one must first modify Protocol 1 so that it would be sender-private if the receiver is semi-honest. If $\mathcal{R}$ is a quasigroup (that is, if $\forall a, b \in \mathcal{R}$ there exist unique $x, y \in \mathcal{R}$ such that $ax = b$ and $ya = b$, then also $x\mathcal{R} = \mathcal{R}$ for any $x \in \mathcal{R}$), then it is sufficient that Sender masks all intermediate values $w_j$ by multiplying them with a random encryption of $0$. Additionally, it is necessary for Receiver to prove the correctness of his public key; this step can be done in a setup phase of the protocol only once per every Sender, after that the same key can be used in many executions of the same protocol. We will assume that Protocol 1 has been modified like that, thus this proof of correctness does not increase the number of rounds. Due to the lack of space we omit the proof that this can be done in a secure way. We omit description of some possible resulting oblivious transfer protocols—based on the Naor-Pinkas transformation [NP99] and on the zero-knowledge proofs—from the proceedings version of this paper. The Aiello-Ishai-Reingold transformation, described next, is superior to the Naor-Pinkas transformation, since the latter only guarantees computational sender-

privacy, and to the transformation based on zero-knowledge proofs since the latter either takes four rounds or works in a non-standard model (that is, either in the random oracle model or in the common reference string model).

Let $\mathcal{M}$ be a plaintext space and $\mathcal{C}$ a ciphertext space, corresponding to some parameter choice of ElGamal public-key cryptosystem. In [AIR01], the authors proposed the next generic transformation of a $\mathrm{CPIR}^n_{\log \sharp \mathcal{C}}$ protocol to an $\mathrm{OT}^n_{\log \sharp \mathcal{M}}$ protocol: Receiver sends an ElGamal encryption $c$ of the query $q$, together with the first message of $\mathrm{CPIR}^n_{\log \sharp \mathcal{C}}$, to Sender. Sender applies the computations, corresponding to the second step of the $\mathsf{AIR}^n_{\sharp \mathcal{M}}$ protocol, with input $c$, to her database, and then the second step of the $\mathrm{CPIR}^n_{\log \sharp \mathcal{C}}$, to the resulting database of ciphertexts. When applied to $\mathsf{LFCPIR}^n_\ell(\log n)$, the resulting $\mathrm{OT}^n_\ell$ protocol has communication $\log \sharp \mathcal{C} + (\frac{\xi}{2} \cdot \log^2 n + (s' + \frac{3\xi}{2} \log n + s')k$ instead of $(\frac{\xi}{2} \cdot \log^2 n + (s + \frac{3\xi}{2}) \log n + s)k$ in the $\mathsf{LFCPIR}^n_{\sharp \mathcal{M}}(\log n)$ protocol. Here, $s$ and $s'$ are the smallest integers, such that $sk \geq \log \sharp \mathcal{M}$ and $s'k \geq \log \sharp \mathcal{C}$; usually $s' = 2s$. Therefore, this transformation increases communication by $\log \sharp \mathcal{C} + (s \cdot \log n + s)k$ bits. The resulting

oblivious transfer protocol is *information-theoretically sender-private* (not like the protocol based on the Naor-Pinkas transform) if ElGamal is IND-CPA secure and $\Pi$ is IND-LFCPA secure, that is, *in the standard complexity-theoretic model* (not like the protocol based on non-interactive honest-verifier zero-knowledge proofs). However, it still makes the additional assumption that ElGamal is IND-CPA secure.

## 5 Comparisons

Fix $k = 1024$ and $s = 1$. The difference between the communications of the linear Aiello-Ishai-Reingold CPIR $\mathsf{AIR}^n_\ell$ [AIR01] (with communication $2(n+1)k$), the polylogarithmic CPIR $\mathsf{CMS}^n_\ell$ [CMS99] (with possibly overly optimistic setting $\kappa = \min(80, \log^2 n)$ and $f = 4$; whether the $\mathsf{CMS}^n_\ell$ CPIR is actually secure in this setting is unknown), the superpolylogarithmic $\mathsf{HomCPIR}^n_\ell(\mathsf{sqrtlog}(2, n))$, and $\mathsf{LFCPIR}^n_\ell(\log n)$ is depicted by Fig. 1. For small values of $\ell$, the best solution is to use the $\mathsf{LFCPIR}^n_\ell(\frac{\ln 2}{W(-2e^{-2})+2} \cdot \log n)$ protocol. For large values of $\ell$, one might to use $\mathsf{LFCPIRBIG}^n_\ell(\alpha)$ with a suitably tuned $\alpha$, say $\alpha = \mathsf{sqrtlog}(2, n)$.

Computation-efficiency is an important property of the $\mathsf{LFCPIR}^n_\ell(\alpha)$ protocol since otherwise in some applications one would prefer a protocol with a smaller computational complexity but with linear communication. Moreover, in practice, Sender's huge computation is mostly likely going to be the first obstacle in applying $\mathrm{CPIR}^n_\ell$ protocols on large databases. In $\mathsf{LFCPIR}^n_\ell(\log n)$, Sender's computation is $\Theta(n\ell)$ $k$-bit exponentiations, which is asymptotically optimal in $n$. This compares favourable with $\Theta(\ell \cdot n 2^{\mathsf{sqrtlog}(\eta, n)})$ $k$-bit exponentiations in $\mathsf{HomCPIR}^n_\ell(\mathsf{sqrtlog}(\eta, n))$. In particular, Sender's computation cost in $\mathsf{LFCPIR}^n_\ell(\log n)$ is comparable to that of the 1-out-of-$n$ oblivious transfer protocols from [NP01,AIR01] that have linear communication.

**Acknowledgements.** We would like to thank Yan-Cheng Chang, Sven Laur and anonymous referees for useful comments. This work was partially supported by the Estonian Science Foundation, grant 6096.

## 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.

[Cha04] Yan-Cheng Chang. Single Database Private Information Retrieval with Logarithmic Communication. In Josef Pieprzyk and Huaxiong Wang, editors, *The 9th Australasian Conference on Information Security and Privacy (ACISP 2004)*, volume 3108 of *Lecture Notes in Computer Science*, pages 50–61, Sydney, Australia, July 13–15, 2004. Springer-Verlag.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computational Private Information Retrieval with Polylogarithmic Communication. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414, Prague, Czech Republic, May 2–6, 1999. 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.

[DJ03] Ivan Damgård and Mads Jurik. A Length-Flexible Threshold Cryptosystem with Applications. In Rei Safavi-Naini, editor, *The 8th Australasian Conference on Information Security and Privacy*, volume 2727 of *Lecture Notes in Computer Science*, pages 350–364, Wollongong, Australia, July 9-11, 2003. Springer-Verlag.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is Not Needed: Single Database, Computationally-Private Information Retrieval. In *38th Annual Symposium on Foundations of Computer Science*, pages 364–373, Miami Beach, Florida, October 20–22, 1997. IEEE Computer Society.

[LAN02] Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure Vickrey Auctions without Threshold Trust. In Matt Blaze, editor, *Financial Cryptography — Sixth International Conference*, volume 2357 of *Lecture Notes in Computer Science*, pages 87–101, Southhampton Beach, Bermuda, March 11–14, 2002. Springer-Verlag.

[Lip04] Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Total Communication. Technical Report 2004/063, International Association for Cryptologic Research, February 25, 2004.

[NP99] Moni Naor and Benny Pinkas. Oblivious Transfer and Polynomial Evaluation. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 245–254, Atlanta, Georgia, USA, May 1–4, 1999. ACM Press.

[NP01] Moni Naor and Benny Pinkas. Efficient Oblivious Transfer Protocols. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 448–457, Washington, DC, USA, January 7–9, 2001. ACM Press.

[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.

[Ste98] Julien P. Stern. A New and Efficient All or Nothing Disclosure of Secrets Protocol. In Kazuo Ohta and Dingyi Pei, editors, *Advances on Cryptology — ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 357–371, Beijing, China, October 18–22, 1998. Springer-Verlag.