An Oblivious Transfer Protocol with Log-Squared Communication

*** Fourth Public Draft. October 12, 2004 ***

Helger Lipmaa

Laboratory for Theoretical CS, Department of CS&E
Helsinki University of Technology, P.O.Box 5400, FIN-02015 HUT, Espoo, Finland
{helger}@tcs.hut.fi

Abstract. We propose a family of one-round 1-out-of-n computationally-private information retrieval protocols for elements from $\mathbb{Z}_d$ that has the next properties: (a) In the asymptotically optimal case, it has communication $\Theta(\log^2 n \cdot k + \log n \cdot \log d)$ bits, where k is the security parameter; (b) It can be based on an arbitrary IND-CPA secure length-flexible public-key cryptosystem. In particular, the sender-privacy of the new protocols can be based on the assumption that the Decisional Composite Residuosity Problem is hard.The proposed protocols can be transformed to one-round computationally chooser-private and information-theoretically sender-private 1-out-of-n oblivious-transfer protocols for elements from $\mathbb{Z}_d$, with the same asymptotic communication, that is secure assuming that the underlying cryptosystem is IND-CPA secure, i.e., in the standard model.
Keywords: computationally-private information retrieval, homomorphic public-key cryptosystem, oblivious transfer.

1 Introduction

During a 1-out-of-n computationally-private information retrieval protocol for elements from $\mathbb{Z}_d$ (CPIR_d^n), Chooser retrieves an entry from Sender's database $\mu = (\mu[1], \dots, \mu[n])$, so that a computationally bounded Sender will not obtain any information on which element was retrieved. A CPIR_d^n protocol is a (computationally chooser-private and information-theoretically sender-private) 1-out-of-n oblivious transfer protocol for elements from d, OT_d^n, if additionally, on Chooser's input σ, Chooser's retrieved information is restricted to the element $\mu[\sigma]$. Both CPIR_d^n and OT_d^n have important cryptographic applications and have therefore been studied by many researchers.

The first CPIR_2^n protocol with polylogarithmic (in n) communication was proposed by Cachin, Micali and Stadler in [CMS99]. The security of the CMS_2^n CPIR_2^n protocol is based on a rather new Φ Assumption, and its total communication is $\kappa^4 + 3\kappa^f = \Omega(\log^8 n + \log^{2f} n)$, where $\kappa > \log^2 n$ is a security parameter and f is a constant in the underlying Φ assumption. We are not aware of any realistic hypothesis about the actual value of this constant except that $f \geq 4$ [CMS99].

Slightly earlier, in 1998, Julien P. Stern [Ste98] proposed a family—that we call $\mathsf{HomCPIR}_d^n(\alpha)$—of CPIR_d^n protocols, based on an arbitrary additively homomorphic IND-CPA secure public-key cryptosystem Π, for transferring elements from the plaintext space of Π. In the asymptotically near-optimal case $\alpha = \log_\eta^{1/2} n$, $\mathsf{HomCPIR}_d^n(\alpha)$ has still super-polylogarithmic communication $(1+o(1))\eta\cdot\log_\eta^{1/2} n\cdot \eta^{\log_\eta^{1/2} n}\cdot k$, where k is the security parameter and $\eta \geq 2$ is the ciphertext expansion ratio of the underlying homomorphic cryptosystem; $k \geq 1024$ and $\eta = 2$ in the case of Paillier's cryptosystem [Pai99]. This is comparable to the asymptotic running time of the quadratic field sieve factorisation algorithm. For all practical values of n, Stern's CPIR is more communication-efficient than the Cachin-Micali-Stadler cryptosystem.

We propose a modification to $\mathsf{HomCPIR}_d^n(\alpha)$ that takes advantage of the concept of length-flexible additively homomorphic cryptosystems [DJ01,DJ03]. Recall that a length-flexible cryptosystem has an additional parameter s, such that given only a single public and private key pair of the receiver and a random value belonging to a fixed s-independent set, the encryption algorithm maps plaintexts from an s-dependent set $\mathcal{M}_s$ to ciphertexts in an s-dependent set $\mathcal{C}_s$, such that $\mathcal{C}_s \subseteq \mathcal{M}_{s+\xi}$ for some small constant ξ. Assume that all database elements are members of some set $\mathbb{Z}_d$, and that $s \in \{1, 2, \dots\}$ is the minimal integer such that $sk > \log d$. Our main result is as follows.

Theorem 1. *Assume $\log d < sk$ for some $s \in \mathbb{Z}^+$. For any $\alpha \in [\log n]$, there exists a CPIR_d^n protocol for elements in $\mathbb{Z}_d$, that we call $\mathsf{LFCPIR}_d^n(\alpha)$, with communication $(\alpha(s + (\alpha+1)\xi/s) \cdot (n^{1/\alpha} - 1) + s + \alpha\xi) \cdot k$ bits. In the asymptotically optimal case $\alpha = \log n$, the communication of $\mathsf{LFCPIR}_d^n(\alpha)$ is $(\frac{\xi}{2} \cdot \log^2 n + (s + \frac{3\xi}{2}) \cdot \log n + s) \cdot k = \Theta(\log^2 n \cdot k + \log n \cdot \log d)$. Moreover, $\mathsf{LFCPIR}_d^n(\alpha_0 \cdot \log n)$ requires from Server work that is equivalent to $\Theta(n)$ sk-bit exponentiations.*

Note that if $\log d \geq \log n \cdot k$, then $\mathsf{LFCPIR}_d^{\log n}(\log n)$ has communication $\Theta(\log n \cdot \log d)$; this is very close to the communication of non-private information retrieval ($\log n + \log d$). Therefore, the basic version of the new CPIR_d^n protocol is especially communication-efficient for longer documents.

With a slight tuning, we can turn $\mathsf{LFCPIR}_d^n(\alpha)$ into a related CPIR_d^n protocol $\mathsf{LFCPIRHR}_d^n(\alpha)$, such that $\mathsf{LFCPIRHR}_d^n(0.371 \cdot \log n)$ has communication $(0.273 \cdot \log n + (0.627 \cdot s + 0.314) \cdot \log\log n + O(1)) \log n \cdot k = \Theta(\log^2 n \cdot k + \log d \cdot \log n \cdot \log\log n)$. The protocol $\mathsf{LFCPIRHR}_d^n(\alpha_0 \cdot \log n)$ is more efficient than $\mathsf{LFCPIR}_d^n(\log n)$ for small values of d. We will propose some more optimisation techniques that are especially efficient for particular values of d or n and that enable to balance communication and computation.

While length-flexible cryptosystems have been used previously say to improve the efficiency of electronic voting [DJ01] and electronic auction protocol [LAN02], all the previous such protocols, that we are aware of, make use of encryptions only under a single—although possible very large—value of s. On the other hand, the transcript of the the proposed CPIR_d^n protocols includes encryptions of interrelated plaintexts under different values of s. This calls for a new security notion for length-flexible cryptosystems, α-IND-LFCPA security. Informally, we say that a cryptosystem is α-IND-LFCPA-secure if the attacker cannot distinguish an α-tuple of encryptions (under α different values of s) of two plaintexts, chosen by herself. By a standard hybrid argument, any IND-CPA-secure cryptosystem is α-IND-LFCPA-secure under α-times security degradation. In addition, we show that in the case of the Damgård-Jurik cryptosystem from the PKC 2001, α-IND-LFCPA security is equivalent to IND-CPA security under a tight reduction. We argue that IND-LFCPA security seems to be a necessary requirement for any length-flexible cryptosystem and therefore one should prove tight IND-LFCPA security for all length-flexible cryptosystems.

We prove the computational chooser-privacy of the $\mathsf{LFCPIR}_d^n(\alpha)$ protocol in the setting of concrete security. In particular, we show that if an adversary A can break the computational chooser-privacy of $\mathsf{LFCPIR}_d^n(\alpha)$ with probability ε and in time t, one can build an algorithm that uses A as an oracle, that breaks the IND-LFCPA security of the underlying cryptosystem with the same probability and time, approximately equal to the sum of t and the time that it takes for a honest chooser to participate in $\mathsf{LFCPIR}_d^n(\alpha)$. In particular, this means that $\mathsf{LFCPIR}_d^n(\alpha)$, when used in conjunction with the Damgård-Jurik cryptosystem from the PKC 2001, is computationally chooser-private under the Decisional Composite Residuosity Assumption [Pai99].

We briefly discuss the potentially stronger setting where security against adversaries, working in time $\mathrm{poly}(n, k)$, is desired. Since 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 the general number field sieve, one must have $k = \Omega(\log^{3-o(1)} n)$. Thus, if security against such adversaries is desired, the protocol $\mathsf{LFCPIR}_d^n(\alpha_0 \log_n)$ has communication $\Omega(\log^{5-o(1)} n + \log n \cdot \log d)$. We however argue that such a strong security requirement is not necessary; in practice, it is sufficient to assume that k is large enough so that violating the chooser-privacy in the CPIR cannot be done with "reasonable" probability in "reasonable" time.

We then look at the possibilities to decrease the communication even further. We show that under two alternative additional assumptions on the used cryptosystem, there exist a CPIR with communication $\Theta(\log n \cdot (k + \log d))$. The first assumption is that an encryption of a small constant in a smaller ciphertext-space can be transferred to an encryption of the same constant in a larger ciphertext-space without knowing the private key. The second, alternative, assumption is that the cryptosystem is ring-homomorphic. We conjecture that such cryptosystems exist—with the first assumption being somewhat more plausible than the second assumption—, though we show that the Damgård-Jurik cryptosystem from PKC 2001 [DJ01] does not satisfy either of the two requirements.

By using the the Naor-Pinkas transformation from [NP99], one can modify $\mathsf{LFCPIR}_d^n(\alpha)$ to a one-round OT_d^n protocol with additional communication overhead of only $6 \log n \cdot \kappa$ bits, where the security parameter κ can be as small as 160. The resulting OT_d^n protocol is computationally chooser-private and computationally server-private, given that the $\mathsf{LFCPIR}_d^n(\alpha)$ protocol is computationally chooser-private and that the used function F is a pseudo-random function. By using zero-knowledge protocols for proving that Chooser's inputs are in the correct range, one can construct a two-round OT_d^n protocol that is computationally chooser-private and information-theoretically server-private in the standard model; this can be transformed into secure one-round protocol in the random-oracle model. The most efficient transformation to an oblivious transfer protocol follows from the recent work of Laur and Lipmaa [LL04]; by using their transformation, we get an information-theoretically sender-private one-round OT_d^n protocol, with an constant times communication overhead, that is secure whenever the underlying cryptosystem is IND-LFCPA secure.

Our results indicate that in the case of the CPIR_d^n protocols, one should emphasise complexity-theoretic notions—like poly-logarithmicity—too much, but instead study the communication of a protocol in a very concrete framework. This is best illustrated by the fact that the only previous polylogarithmic CPIR_d^n protocol by Cachin, Micali and Stadler requires more communication than say the Aiello-Ishai-Reingold linear CPIR_d^n protocol AIR_d^n [AIR01] for $n \le 2^{30}$. At the same time, 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 CPIR_d^n protocols for practically any n ("good in practice").

Road-map. In Section 2, we introduce the reader to notation and preliminaries. In Section 4, we propose the new CPIR protocol, prove its security, propose some optimisations and show that it can be efficiently transformed to an oblivious transfer protocol. In Section 5, we propose two different hypothetical CPIR_d^n protocols with communication $\Theta(\log n \cdot (k + \log d)$, and study their (im)possibility. In Section 6, we compare the different CPIR protocols from the efficiency viewpoint and discuss possible extensions.

2 Preliminaries

Notation. For a natural number t, let $[t]$ denote the set $\{1, \dots, t\}$. Let $\mathsf{Eq}(i, j) = 1$ if $i = j$ and 0, otherwise. All logarithms in this paper will be on base 2, unless explicitly mentioned. Let e be the base of the natural logarithm, i.e., $\ln e = 1$. Let W be Lambert's W function, i.e., W satisfies the functional identity $W(x)e^{W(x)} = x$. 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. The statistical difference of two distributions X and Y over the discrete support U is defined as $\mathsf{StatDiff}\,(X \approx Y) := \max_{S \subseteq U} |\Pr[X \in S] - \Pr[Y \in S]|$. Let k and κ be two security parameters, where usually k corresponds to the super-polynomial security (breaking some primitive is hard in time $\mathrm{poly}(k)$) and κ 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 let n to denote Sender's database size.

Decisional composite residuosity problem [Pai99]. Assume that $M = pq$ 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* (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 the general number field sieve.

Homomorphic public-key cryptosystems. Public-key cryptosystem is a triple $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where (a) Gen is the key generation algorithm—that on input 1^k, returns $(\mathrm{SK}, \mathrm{PK})$, where SK is a secret key and PK is a public key—, (b) Enc is the encryption algorithm, that on input $(\mathrm{PK}, m; r)$, where PK is a public key, m is the plaintext and r is the random coin, returns the ciphertext $\mathsf{Enc}_{\mathsf{PK}}(m; r)$, and (c) Dec is the decryption algorithm that on input c returns the corresponding plaintext $m = \mathsf{Dec}_{\mathsf{SK}}(c)$. For fixed Π and public key PK, let $\mathcal{C} = \mathcal{C}(\Pi, \mathrm{PK})$ be the ciphertext space, let $\mathcal{R} = \mathcal{R}(\Pi, \mathrm{PK})$ be the randomness space and let $\mathcal{M} = \mathcal{M}(\Pi, \mathrm{PK})$ be the plaintext space.

A cryptosystem Π is *additively homomorphic* if for any key pair $(\mathrm{SK}, \mathrm{PK})$, any $m, m' \in \mathcal{M} = \mathbb{Z}_{\sharp\mathcal{M}}$ and any $r, r' \in \mathcal{R}$, $(\mathsf{Enc}_{\mathsf{PK}}(m; r) \cdot \mathsf{Enc}_{\mathsf{PK}}(m'; r') = \mathsf{Enc}_{\mathsf{PK}}(m + m'; r \circ r')$, where $\circ$ is a groupoid operation in $\mathcal{R}$, and $+$ is addition in $\mathbb{Z}_{\sharp\mathcal{M}}$. A few homomorphic cryptosystems [OU98,NS98,Pai99,DJ01,DJ03] have been proven to be secure in the sense of IND-CPA under reasonable complexity assumptions.

Let $\eta := \lceil \log \sharp\mathcal{C}/k \rceil$ be the *ciphertext expansion ratio* of Π; $\eta = 2$ for the Paillier cryptosystem [Pai99] and the Damgård-Jurik cryptosystem from PKC 2001 [DJ01], $\eta = 3$ for the Okamoto-Uchiyama cryptosystem [OU98] and $\eta \in \{2,3\}$ for another cryptosystem by Damgård and Jurik [DJ03].

Length-flexible cryptosystems. A length-flexible public-key cryptosystem is a tuple $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where again Gen is the key generation algorithm, Enc is the encryption algorithm and Dec is the decryption algorithm. It is required that for any $(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Gen}(1^k)$ and for any integer $s \geq 1$, $\mathsf{Enc}_{\mathrm{PK}} : \mathcal{M}_s \times \mathcal{R} \to \mathcal{C}_s$, where $\mathcal{C}_s = \mathcal{C}_s(\Pi, \mathrm{PK})$ is the ciphertext space and $\mathcal{M}_s = \mathcal{M}_s(\Pi, \mathrm{PK})$ is the plaintext space corresponding to s, and $\mathcal{R} = \mathcal{R}(\Pi, \mathrm{PK})$ is the randomness space. We require that for some constant a, $\mathcal{C}_s \subseteq \mathcal{M}_{s+a}$ for every s; we assume that $\xi = \xi(\Pi)$ is the minimal among such a's.

Throughout this paper, we assume that $k = \log \sharp\mathcal{M}_1$ is the security parameter. (See Section 6.1 for an additional comment on that.) For the sake of simplicity, we will assume that$\sharp\mathcal{M}_s = (\sharp\mathcal{M}_1)^s$ with $\log \sharp\mathcal{M}_s = sk$, and that $\sharp\mathcal{C}_s = \sharp\mathcal{M}_{s+\xi}$.

Length-flexible cryptosystems have been used before to implement multi-candidate electronic voting [DJ01,DJ03] and large-scale electronic auctions [LAN02] over large plaintext spaces.

Damgård-Jurik cryptosystem $\mathsf{DJ01}$ **from PKC 2001 [DJ01].** The $\mathsf{DJ01}$ cryptosystem was the first published length-flexible cryptosystem. For fixed s, $\mathcal{M}_s := \mathbb{Z}_{M^s}$, $\mathcal{R} = \mathbb{Z}_M^*$ and $\mathcal{C}_s := \mathbb{Z}_{M^{s+1}}^*$, thus $\log \sharp\mathcal{C}_s / \log \sharp\mathcal{M}_s \approx 1 + 1/s$, $\xi = 1$ and $\eta = 2$. Encryption is defined as $\mathsf{Enc}^s_{\mathrm{PK}}(m; r) := (1+M)^m \cdot r^{M^s} \mod M^{s+1}$, where r is randomly chosen from $\mathbb{Z}_M$. (To achieve formal completeness, one can require that r is regenerated if a r with $\gcd(r, M) > 1$ is chosen.) To decrypt $c = \mathsf{Enc}^s_{\mathrm{PK}}(m; r)$, one first computes $c \mod M = r^{M^s} \mod M$, and then retrieves r and thus also $c' = (1+M)^m \mod M^{s+1}$. After that, m is retrieved from c' by using an algorithm from [DJ01] that works in time $O(s^4 k^2)$.

The $\mathsf{DJ01}$ cryptosystem is additively homomorphic since $\mathsf{Enc}^s_{\mathrm{PK}}(m_1; r_1) \cdot \mathsf{Enc}^s_{\mathrm{PK}}(m_2; r_2) = \mathsf{Enc}^s_{\mathrm{PK}}(m_1 + m_2; r_1 r_2)$. Importantly, the same key PK can be used to encrypt messages of different length, and $\mathcal{C}_s \subseteq \mathcal{M}_{s+1}$. For a fixed Π, denote encryption and decryption with key pair $(\mathrm{SK}, \mathrm{PK})$ and a fixed s by $\mathsf{Enc}^s_{\mathrm{PK}}$ and $\mathsf{Dec}^s_{\mathrm{SK}}$.

Damgård-Jurik cryptosystem $\mathsf{DJ03}$ **from ACISP 2003 [DJ03]**. The $\mathsf{DJ03}$ length-flexible cryptosystem is slightly less efficient with $\log \sharp\mathcal{C}_s / \log \sharp\mathcal{M}_s \approx 1 + 2/s$, i.e., it has $\xi = 2$. In this cryptosystem, SK is a k-bit RSA modulus $M = pq$, where p and q are safe primes, g is a quadratic residue modulo M, $\alpha \leftarrow \mathbb{Z}_{2^{k/2}}$. The public key is (M, g, h) where $h = g^\alpha \mod M$, the private key is α (the values of (p, q) can be deleted). Define $\mathcal{M}_s = \mathbb{Z}_{M^s}$, $\mathcal{R} = \mathbb{Z}_{2^{k/2}}$ and $\mathcal{C}_s = \mathbb{Z}_M \times \mathbb{Z}_{M^{s+1}}$. Define $\mathsf{Enc}^s_{\mathrm{PK}}(m; r) = (g^r \mod M, (1+M)^m (h^r \mod M)^{M^s} \mod M^{s+1})$. See [DJ03] for a description of the decryption algorithm.

IND-CPA secure length-flexible public-key cryptosystems. Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a length-flexible public-key cryptosystem. We define the success of a randomised algorithm A in breaking it as follows:

$$\mathsf{Adv}^{\mathsf{indcpa}}_{\Pi,k}(A) := \Big|\Pr[(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Gen}(1^k), (m_0, m_1, s) \leftarrow A(\mathrm{PK}),$$
$$b \leftarrow \{0,1\} : A(\mathrm{PK}, m_0, m_1, s, \mathsf{Enc}^s_{\mathrm{PK}}(m_b; \mathcal{R})) = b] - \frac{1}{2}\Big| \enspace .$$

Here, probability is taken over random coin tosses of Gen, A, $\mathsf{Enc}^s_{\mathrm{PK}}$ and over the choice of b. We say that Π is (ε, t)*-secure in the sense of IND-CPA* if $\mathsf{Adv}^{\mathsf{indcpa}}_{\Pi,k}(A) \leq \varepsilon$ for any probabilistic algorithm A that works in time t. If $t(k)$ is polynomial in k and $\varepsilon(k)$ is negligible in k, we will just say that Π is secure in the sense of IND-CPA.

The $\mathsf{DJ01}$ length-flexible cryptosystem is secure in the sense of IND-CPA, assuming that the DCRP is hard [DJ01], while the $\mathsf{DJ03}$ length-flexible cryptosystem is IND-CPA secure if both the DCRP and on the Decisional Diffie-Hellman Problem are hard [DJ03]. A modified version of the $\mathsf{DJ03}$ cryptosystem is secure under the DCRP assumption. In this modified version, $\hat{g} \leftarrow g^{M^s} \mod M^{s+1}$, $\hat{h} \leftarrow h^{M^s} \mod M^{s+1}$, and $\mathsf{Enc}^s_{\mathrm{PK}}(m; r) = (\hat{g}^r \mod M^{s+1}, (1+M)^m \hat{h}^r \mod M^{s+1})$. (However, then $\mathcal{C}_s \subseteq \mathcal{M}_{2s+2}$ and thus $\xi = s + 2$ is not a constant.)

Computationally-private information retrieval. During a (single-server) 1*-out-of-*n *computationally-private information retrieval* (CPIR^n_d) protocol for elements from $\mathbb{Z}_d$, Chooser fetches $\mu[\sigma]$ from the database $\mu = (\mu[1], \ldots, \mu[n])$, $\mu[i] \in \mathbb{Z}_d$ for some fixed d, so that a computationally bounded Sender does not know which entry Chooser is learning. The communication complexity of information retrieval (without any privacy requirements and no additional information on the structure of the data) is $\lceil \log n \rceil + \lceil \log d \rceil$.

More formally, a one-round computationally-private CPIR_d^n protocol is defined as follows. (Since all protocols, considered in this paper, have one round (that is, two messages: one from Chooser to Sender, and another one from Sender to Chooser), we will not give a general definition.) A CPIR_d^n protocol Γ is a quadruple of algorithms, $(\mathsf{Init}, \mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$, corresponding to the initialisation stage, two messages of the protocol and the recover phase. Here, Init, Query and $\mathsf{Transfer}$ are randomised and $\mathsf{Recover}$ is deterministic. Let $\mathcal{Q}$ and $\mathcal{T}$ be two distributions.

In the initialisation phase, Chooser Cho generates his secret and public key pair $(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Init}(1^k)$. The public key PK is sent to the Sender before the start of the protocol; the same PK can be used in several protocol runs. The first message, $\mathsf{msg}_1 \leftarrow \mathsf{Query}_{\mathrm{PK}}(\sigma, n; r)$, of a protocol run is by Chooser Cho, where σ is his input index, n is the database size and $r \leftarrow \mathcal{Q}$ is a new random value. We assume that Query does not depend on SK; this again holds for all protocols that we are interested in. (If Query depends on SK, latter the security definition of a CPIR_d^n must allow the adversary to access Query as an oracle.) The second message is by Sen, who replies with $\mathsf{msg}_2 \leftarrow \mathsf{Transfer}_{\mathrm{PK}}(\mu, \mathsf{msg}_1; r)$, where μ is her input, $r \leftarrow \mathcal{T}$ is a new random value and msg_1 is the first message of the protocol. (If not interested in the server-privacy, one may usually assume that $\mathsf{Transfer}$ is a deterministic algorithm.) After the second message, Chooser returns his private output $\mathsf{Recover}_{\mathsf{SK}}(\sigma, \mathsf{msg}_1, \mathsf{msg}_2)$.

A CPIR_d^n protocol $\Gamma = (\mathsf{Init}, \mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$ is γ-*correct* if

$$\Pr\big[(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Init}(1^k), \mathsf{msg}_1 \leftarrow \mathsf{Query}_{\mathrm{PK}}(\sigma, n; \mathcal{Q}) : \\ \mathsf{Recover}_{\mathsf{SK}}(\sigma, \mathsf{msg}_1, \mathsf{Transfer}_{\mathrm{PK}}(\mu, \mathsf{msg}_1; \mathcal{T})) = \mu[\sigma]\big] = 1 - \gamma(k)$$

for any k and any $\sigma \in [n]$. The probability is taken over the coin tosses of Init, Query and $\mathsf{Transfer}$. If $\gamma = 0$ then we just say that the protocol is correct.

For a randomised algorithm A executing Sender's part in the CPIR_d^n protocol Γ, define

$$\mathsf{Adv}^{\mathsf{cpir}}_{\mathsf{Cho},k}(A) := \Big|\Pr\big[(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Init}(1^k), (\sigma_0, \sigma_1, n) \leftarrow A(\mathrm{PK}), \overline{b} \leftarrow [0,1] : \\ A(\mathrm{PK}, \sigma_0, \sigma_1, n, \mathsf{Query}_{\mathrm{PK}}(\sigma_{\overline{b}}, n; \mathcal{Q})) = \overline{b}\big] - \frac{1}{2}\Big|$$

to be the advantage over random coin-tossing that A has in guessing, which of the two possible choices σ_0 and σ_1 was used by the chooser, after observing a single query from Chooser. The probability is taken over the coin tosses of Init, A, Query and over the choice of $\overline{b}$. We call Γ a (t, ε)-*secure* CPIR_d^n *protocol*, if $\mathsf{Adv}^{\mathsf{cpir}}_{\mathsf{Cho},k}(A) \leq \varepsilon = \varepsilon(k)$ for any probabilistic algorithm A that works in time $t = t(k)$. In Section 4, we will give a longer comment on the necessity of talking in the terms of the concrete security, and why t and ε should not depend on n.

The first CPIR_d^n with sub-linear communication $O(2^{\log^{1/2} n \cdot \log^{1/2} k})$ was proposed by Kushilevitz and Ostrovsky in [KO97]. The first CPIR_1^n with polylogarithmic total communication was proposed by Cachin, Micali and Stadler in [CMS99]. However, the security of the CMS_2^n CPIR protocol is based on the relatively little-studied Φ Assumption that basically states that given a "large" composite M with unknown factorisation and a "small" prime p, it is hard to decide whether $p \mid \phi(M)$. Moreover, the CMS_2^n protocol has total communication $\kappa^4 + 3\kappa^f = \Omega(\log^8 n + \log^{2f} n)$ for some constant f and a security parameter $\kappa > \log^2 n$. In particular, its communication depends on a constant f, existence of which is conjectured by the Φ Assumption. No hypothesis about the value of this constant 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 $p \mid \phi(m)$.

The Cachin-Micali-Stadler CPIR_1^n protocol retrieves one bit per every run, and we are not aware of any more efficient to transform it to a CPIR_d^n protocol than to run it $\lceil \log d \rceil$ times in parallel (with the same Chooser's query); this means that the CMS_d^n CPIR_d^n protocol has communication $\Omega((\log^{2f} n \cdot \log d + \log^8 n)$. Even if polylogarithmic, the communication of the CMS_d^n protocol is larger than the communication of some linear CPIR protocols [AIR01] for all relevant database sizes. (See Fig. 1.)

Stern's CPIR_d^n. The Kushilevitz-Ostrovsky CPIR_d^n was first generalised by Julien P. Stern [Ste98] and then further refined by Chang [Cha04]. While the security of the Kushilevitz-Ostrovsky CPIR_d^n is based on the quadratic residuosity assumption and enables the chooser to retrieve with every query a single bit, Stern's CPIR_d^n is based on an arbitrary IND-CPA secure additively homomorphic cryptosystem Π, and enables the chooser to retrieve up to $k = \log \sharp\mathcal{M}$ bits with every query. We next give a self-contained description of Stern's CPIR_d^n. For this, we recast Stern's CPIR_d^n in the contemporary language of homomorphic cryptosystems, and by using the hypercube-based description from [Cha04].

Stern's CPIR_d^n is a family of CPIR-s, indexed by the parameter $\alpha \geq 1$. We call this family $\mathsf{HomCPIR}_d^n(\alpha)$. The main idea behind $\mathsf{HomCPIR}_d^n(\alpha)$ is as follows. Look at the database μ as at the α-dimensional hypercube with its elements $\mu(i_1, \dots, i_\alpha)$ addressed by coordinates $(i_1, \dots, i_\alpha)$; Chooser's query σ has thus also α coordinates, $\sigma = (\sigma_1, \dots, \sigma_\alpha)$. For every dimension i and every coordinate j along the ith dimension, Chooser sends an encryption $\beta_{i,j}$ of $\mathsf{Eq}(\sigma_i, j)$ to Sender. Fix a dimension, say the 1st. Sender constructs an $(\alpha - 1)$-dimensional database as follows. For all dimensions $j \neq 1$ and for all coordinates i_j along those dimensions, set the $(i_2, \dots, i_j)$th element of the new database to be equal to the product $\prod_t \beta_{1,t}^{\mu(t,i_2,\dots,i_\alpha)}$. Due to the homomorphic properties of Π, this product is an encryption of $\mu(\sigma_1, i_2, \dots, i_\alpha)$. The same process is repeated with the $(\alpha - 1)$-dimensional database of encrypted values $\mathsf{Enc}_{\mathrm{PK}}(\mu(\sigma_1, i_2, \dots, i_\alpha))$, with the $(\alpha - 2)$-dimensional database of doubly-encrypted values $\mathsf{Enc}_{\mathrm{PK}}(\mathsf{Enc}_{\mathrm{PK}}(\mu(\sigma_1, \sigma_2, i_3, \dots, i_\alpha)))$, until Sender obtains an α-times encryption of $\mu(\sigma_1, \dots, \sigma_\alpha)$. Sender sends this value to the Chooser, who decrypts it α times.

Now, $\mathsf{HomCPIR}_d^n(\alpha)$ deviates from this basic idea since Π must have ciphertext expansion ratio $\eta \geq 2$. To account with this, for any j, $\mathsf{HomCPIR}_d^n(\alpha)$ creates η^j $(\alpha - j)$-dimensional intermediate databases of j-times encrypted ciphertexts. Sender transfers η^α α-times encrypted ciphertexts to Chooser, who recursively combines their decryptions to obtain $\mu(\sigma_1, \dots, \sigma_\alpha)$. Thus, $\mathsf{HomCPIR}_d^n(\alpha)$ has chooser-side communication $\eta\alpha n^{1/\alpha} k$ bits and sender-side communication $\eta^\alpha k$ bits. One can minimise the communication by optimising the value of α. However, even in the close-to-optimal case when $\alpha = \log_\eta^{1/2} n$, the communication $\Theta(\eta \cdot \log_\eta^{1/2} n \cdot \eta^{\log_\eta^{1/2} n}) \cdot k$ is still super-logarithmic. Here, $\alpha = \log_\eta^{1/2} n$ is the "on limit" solution of the equation $\alpha n^{1/\alpha} = \eta^\alpha$. ([Ste98] erroneously claims that the communication of $\mathsf{HomCPIR}_d^n(\log_\eta^{1/2} n)$ is $\Theta(\eta^{\log_\eta^{1/2} n}) \cdot k$.) Thus,

Proposition 1 ([Ste98,Cha04]). *Let Π be a IND-CPA secure homomorphic cryptosystem. (a) Let $\alpha \in [n]$. W.l.o.g., assume that Sender's database $\mu = (\mu[1], \dots, \mu[n])$ contains $n = \lambda^\alpha$ entries of bit-length $\leq k$. There exists an CPIR_d^n protocol $\mathsf{HomCPIR}_d^n(\alpha)$ with chooser-side communication $\alpha n^{1/\alpha} \eta k$ and sender-side communication $\eta^\alpha k$ bits. (b) Fix $\alpha = \log_\eta^{1/2} n$. There exists an CPIR_d^n protocol $\mathsf{HomCPIR}_d^n(\log_\eta^{1/2} n)$ with the communication $(\eta \cdot \log_\eta^{1/2} n + 1) \cdot \eta^{\log_\eta^{1/2} n} \cdot k$ bits.*

Compared with the CMS_d^n protocol, Stern's protocol is significantly more communication-efficient for all relevant database sizes $n \leq 2^{80}$. (See Fig. 1.)

Oblivious transfer. A CPIR_d^n protocol $(\mathsf{Init}, \mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$ is an *(computationally chooser-private and information-theoretically sender-private) 1-out-of-n oblivious transfer protocol for elements from $\mathbb{Z}_d$* (an OT_d^n protocol) if also Sender's privacy is guaranteed. For the formal definition, we make the comparison to the ideal implementation, using a trusted third party that receives μ from Sender, receives σ from Chooser, and sends $\mu[\sigma]$ to Chooser. We assume that Chooser receives garbage (i.e., a random value from some μ-independent set T) if $\sigma \notin [n]$. (Alternatively, one could require that Sender halts if $\sigma \notin [n]$.)

We define the sender-privacy of an OT_d^n protocol by requiring that for every algorithm A, one can define a simulator S that, given only the private input σ, random tape r_A, and private output $\mu[\sigma]$ of A, generates output that is statistically indistinguishable from the view of A that reacts with the honest sender Sen. More precisely, for a honest sender Sen and an algorithm S, define

$$\mathsf{Adv}^{\mathsf{otsen}}_{\mathsf{Sen},k}(A, S) := \min_{\mathrm{PK}} \mathsf{StatDiff}\left(S(1^k, \mathrm{PK}, \sigma, r_A, \mu[\sigma]) \approx (\mathsf{msg}_1, \mathsf{msg}_2)\right) \quad ,$$

where $\mathsf{msg}_1 := A(\mathrm{PK}, \sigma)$, $\mathsf{msg}_2 = \mathsf{Transfer}_{\mathrm{PK}}(\mu, \mathsf{msg}_1; \mathcal{T})$ and PK is a legitimate public key, generated by $\mathsf{Init}(1^k)$.

We say that a (t, ε)-secure CPIR_d^n protocol $\Gamma = (\mathsf{Init}, \mathsf{Query}, \mathsf{Transfer}, \mathsf{Recover})$ is a *computationally* (t, ε)*-chooser-private and statistically* ε'*-sender-private* 1*-out-of-*n *oblivious transfer protocol* if for every not algorithm A, there exists a simulator S, such that $\mathsf{Adv}^{\mathsf{otsen}}_{\mathsf{Sen},k}(A, S) \le \varepsilon'$. As usually, sender-privacy is said to be perfect if $\mathsf{Adv}^{\mathsf{otsen}}_{\mathsf{Sen},k}(A, S) = 0$.

As argued, e.g., in [NP01, Section 2.1.2], an oblivious transfer protocol does not have to guarantee the correctness (even if Cho is honest but Sen is not, Cho will still receive Sen's input $\mu[\sigma]$). Following this convention, also we will leave it up to the application protocols to provide security in this sense.

Naor-Pinkas transformation. An arbitrary CPIR_d^n protocol can be transferred to a OT_d^n protocol by using the methodology of Naor and Pinkas [NP99]. In particular, it can be applied to the Cachin-Micali-Stadler and Stern's CPIR_d^n protocols.

Assume for the sake of simplicity that $n = 2^\lambda$. Let $F : \{0,1\}^t \times \mathbb{Z}_d \to \mathbb{Z}_d$ be a pseudo-random function. Sender prepares λ random pairs of keys $\{(K_i^0, K_i^1)\}$, $K_i^b \in \{0,1\}^t$, for $i \in [\lambda]$. Assume that $\mu'[i] := \mu[i] \oplus \bigoplus_{j=1}^{\lambda} F(K_j^{i_j}, i)$, where $(i_1, \dots, i_\lambda)$ are the bits of i. Chooser and Sender execute $\mathrm{OT}_{2^t}^2$ for each $1 \le j \le \lambda$ on the strings (K_j^0, K_j^1), with query σ_j. Chooser executes CPIR_d^n to the database $\mu' = (\mu'[1], \dots, \mu'[n])$ and recovers $\mu'[\sigma]$. He recovers $\mu[\sigma] \leftarrow \mu'[\sigma] \oplus \bigoplus_{j=1}^{\lambda} F(K_j^{\sigma_j}, \sigma)$.

Thus, for this OT_d^n protocol, one must in parallel execute $\log n$ times an $\mathrm{OT}_{2^t}^2$ protocol and one time a CPIR_d^n protocol. As shown in [AIR01], there exists a one-round $\mathrm{OT}_{2^t}^2$ protocol $\mathsf{AIR}_{2^t}^2$ (identical to their $\mathrm{CPIR}_{2^t}^2$ protocol), based on the IND-CPA security of the ElGamal cryptosystem, with communication $6 \cdot \kappa$ bits, where the security parameter κ can be as small as 160. Thus, this methodology increases the communication only by $6 \log n \cdot \kappa$ bits.

3 On IND-LFCPA Secure Length-Flexible Cryptosystems

In the CPIR_d^n protocols, proposed in the next sections, we have the next novel adversarial situation. Given a length-flexible cryptosystem $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, the adversary obtains encryptions of interrelated values but by using the different value of s—where s is, again, 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 a generally desirable security feature of length-flexible cryptosystems whenever it might be the case that the adversary obtains different-length encryptions of related plaintexts. That is, 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 length-flexible cryptosystems are secure also in such an adversarial situation.

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a length-flexible public-key cryptosystem. We define the success of a randomised algorithm A in breaking Π's IND-LFCPA security as follows:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{lfindcpa}}_{\Pi,k}(A, \alpha) := \Big|\Pr[&(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Gen}(1^k), (m_0, m_1, s_1, \dots, s_\alpha) \leftarrow A(\mathrm{PK}),\\ &b \leftarrow \{0,1\}, c_1 \leftarrow \mathsf{Enc}^{s_1}_{\mathrm{PK}}(m_b; \mathcal{R}), \dots, c_\alpha \leftarrow \mathsf{Enc}^{s_\alpha}_{\mathrm{PK}}(m_b; \mathcal{R}) :\\ &A(\mathrm{PK}, m_0, m_1, s_1, \dots, s_\alpha, c_1, \dots, c_\alpha) = b] - \frac{1}{2}\Big| \enspace .\end{aligned}$$

Here, probability is taken over random coin tosses of Gen, $\mathsf{Enc}^{(\cdot)}_{\mathrm{PK}}$, A and over the choice of b. We say that Π is (ε, t)*-secure in the sense of* α*-IND-LFCPA* if $\mathsf{Adv}^{\mathsf{lfindcpa}}_{\Pi,k}(A, \alpha) \le \varepsilon$ for any probabilistic algorithm A that works in time t. If $t(k)$ is polynomial in k and $\varepsilon(k)$ is negligible in k, we will just say that Π is secure in the sense of α-IND-LFCPA. We omit α when α can be any polynomial in k.

By a standard hybrid argument, $(\alpha\varepsilon, t - O(\alpha))$-security in the sense of α-IND-LFCPA follows from the (ε, t)-security in the sense of IND-CPA. However, since IND-LFCPA security is such a basic notion for length-flexible cryptosystems, it makes sense to not first prove that a cryptosystem is IND-CPA secure and then use this hybrid argument, but to try to prove IND-LFCPA security directly, without this α-times security degradation.

We will next show that for both known length-flexible cryptosystems ($\mathsf{DJ01}$ and $\mathsf{DJ03}$), IND-LFCPA security follows from IND-CPA security under a tight reduction. First, we give the next lemma. It is motivated by the observation that IND-LFCPA is a potentially stronger security notion than IND-CPA 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 parameter s.

Lemma 1. *Assume that Π is a length-flexible cryptosystem that is (ε, t)-secure in the sense of IND-CPA. Assume there exists an algorithm Z, such that for all $(SK, PK) \leftarrow \mathsf{Gen}(1^k)$ any $s_1 < s_2$, and any $m \in \mathcal{M}_{s_1}$, $Z(PK, s_1, s_2, \mathsf{Enc}^{s_2}_{PK}(m;\mathcal{R})) = \mathsf{Enc}^{s_1}_{PK}(m;\mathcal{R})$, and that can be computed in time $t_Z(1^k, s_2)$. Then, Π is $(\varepsilon, t - \alpha \cdot t_Z(1^k, s_2) - O(\alpha))$-secure in the sense of α-IND-LFCPA.*

Proof. Really, assume A is an adversary who breaks the α-IND-LFCPA security in time t' and with probability ε. Construct the next adversary M^A that breaks the IND-CPA security of Π: Obtain from the encryption black box a new random public key PK, send this to A. M asks A to produce $(m_0, m_1, s_1, \dots, s_\alpha)$. Assume that $s_1 < s_2 < \dots < s_\alpha$. Give (m_0, m_1, s_α) to the black box, who returns $c_\alpha \leftarrow \mathsf{Enc}^{s_\alpha}_{\mathsf{PK}}(m_b;\mathcal{R})$. Compute $c_i \leftarrow Z(\mathrm{PK}, s_i, s_\alpha, \mathsf{Enc}^{s_2}_{\mathsf{PK}}(m;\mathcal{R}))$ for $i \in [\alpha - 1]$. Send $(c_1, \dots, c_\alpha)$ 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 Z. In the case of the DJ01 cryptosystem, $\mathsf{Enc}^{s}_{\mathsf{PK}}(m;r) = (1+M)^m r^{M^s} \mod M^{s+1}$. Therefore, $\mathsf{Enc}^{s_1}_{\mathsf{PK}}(m;\mathcal{R}) = \mathsf{Enc}^{s_2}_{\mathsf{PK}}(m;r) \mod M^s) \cdot \mathsf{Enc}^{s_1}_{\mathsf{PK}}(0;\mathcal{R})$. In the case of the DJ03 cryptosystem, $\mathsf{Enc}^{s}_{\mathsf{PK}}(m;r) = (g^r \mod M, (1+M)^m (h^r \mod M)^{M^s} \mod M^{s+1})$. Therefore, given $\mathsf{Enc}^{s_2}_{\mathsf{PK}}(m;r) = (a, b)$, one can compute $\mathsf{Enc}^{s_1}_{\mathsf{PK}}(m;\mathcal{R}) = (a, b \mod M^{s_1}) \cdot \mathsf{Enc}^{s_1}_{\mathsf{PK}}(0;\mathcal{R})$.

We would get a similar security result, if there existed an efficient function Z', such that for $s_2 < s_1$, and for any $m \in \mathcal{M}_{s_2}$, $Z'(\mathrm{PK}, s_1, s_2, \mathsf{Enc}^{s_2}_{\mathsf{PK}}(m;\mathcal{R})) = \mathsf{Enc}^{s_1}_{\mathsf{PK}}(m;\mathcal{R})$. However, as we show in Section 5, the existence of such a function would additionally result in a CPIR^n_d protocol with logarithmic communication.

4 New CPIR^n_d with Log-Squared Communication

4.1 Basic Construction and Security Proof

Next, we use a length-flexible additively homomorphic public-key cryptosystem to improve both the concrete and the asymptotic total communication of $\mathsf{HomCPIR}^n_d(\alpha)$, by presenting a family $\mathsf{LFCPIR}^n_d(\alpha)$ (*length-flexible computationally-private information retrieval*) of CPIR^n_d protocols.

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a length-flexible cryptosystem. Assume that $\mathbb{Z}_d \subseteq \mathcal{M}_s$ for some fixed s. That is, $\log d < sk = s \log M$ for some public key M. Fix $\alpha \geq 1$. In realistic settings, $\alpha \in [\log n]$. Consider the database $\mu = (\mu[1], \dots, \mu[n])$ as an α-dimensional $\lambda_1 \times \dots \times \lambda_\alpha$ hyper-rectangle, for some λ_i defined later. We assume that $n = \prod_{i=1}^{\alpha} \lambda_i$ and define $\mu(i_1, \dots, i_\alpha) := \mu[(i_1 - 1)\prod_{i=2}^{\alpha} \lambda_i + (i_2 - 1)\prod_{i=3}^{\alpha} \lambda_i + \dots + (i_{\alpha-1} - 1)\lambda_\alpha + (i_\alpha - 1) + 1]$ for $i_j \in [\lambda_j]$.

Thus, every element of μ is indexed by a tuple $(i_1, \dots, i_\alpha)$. If $\alpha > 1$, Sender computes an $(\alpha-1)$-dimensional database $w_1 \in \mathcal{M}_{s+\xi}^{\prod_{i=2}^{\alpha} \lambda_i}$ as follows: for every $(i_2, \dots, i_\alpha)$, let $w_1(i_2, \dots, i_\alpha) \leftarrow \prod_{t \in [\lambda_1]} \beta_{1,t}^{\mu(t, i_2, \dots, i_\alpha)}$, where $\mathsf{Dec}^{s}_{\mathsf{SK}}(\beta_{1,t}) = \mathsf{Eq}(\sigma_1, t)$, i.e., $\beta_{1,t}$ is an encryption of 1 if $\sigma_1 = t$ and an encryption of 0, otherwise. Here, $\beta_{1,t}$ is obtained from Chooser, except in the case $t = \alpha_i$ since then Sender can compute it herself, based on the observation that $\sum_{t=1}^{\lambda_i} \mathsf{Eq}(\sigma_i, t) = 1$. Thus, $\mathsf{Dec}^{s}_{\mathsf{SK}}(w_1(i_2, \dots, i_\alpha)) = \mu(\sigma_1, i_2, \dots, i_\alpha)$. If $\alpha > 2$, Sender repeats this process recursively on the database w_1 to obtain an $(\alpha-2)$-dimensional database $w_2 \in \mathcal{M}_{s+2\xi}^{\prod_{i=3}^{\alpha} \lambda_i}$, such that $\mathsf{Dec}^{s+\xi}_{\mathsf{SK}}(w_2(i_3, \dots, i_\alpha)) = w_1(\sigma_2, i_3, \dots, i_\alpha)$, then—if $\alpha > 3$—an $(\alpha-3)$-dimensional database $w_3 \in \mathcal{M}_{s+3\xi}^{\prod_{i=4}^{\alpha} \lambda_i}$, such that $\mathsf{Dec}^{s+2\xi}_{\mathsf{SK}}(w_3(i_4, \dots, i_\alpha)) = w_2(\sigma_3, i_4, \dots, i_\alpha)$, etc.

Thus, the basic idea of the new protocol is similar to that of $\mathsf{HomCPIR}^n_d(\alpha)$, as described in Section 2. After α steps, Sender has a single element $w_\alpha \in \mathcal{M}_{s+\alpha\xi}$, an α-times encrypted $\mu(\sigma_1, \dots, \sigma_\alpha)$. He then returns w_α to Chooser, who decrypts it α times to obtain $\mu(\sigma_1, \dots, \sigma_\alpha)$. Since the Damgård-Jurik cryptosystem has flexible-length plaintexts, instead of dividing every intermediate ciphertext into η chunks (as in the case of Stern's CPIR^n_d), we just increase the length of the plaintexts. Therefore, it suffices for Sender to just send one value w_α, with length $|w_\alpha| = (s + \alpha\xi)k$, to Chooser.

PRIVATE INPUT: Chooser has $\sigma = (\sigma_1, \dots, \sigma_\alpha)$, Sender has μ.
PRIVATE OUTPUT: Chooser has $\mu(\sigma_1, \dots, \sigma_\alpha)$.

Chooser, $\mathsf{Init}(1^k)$**:**
Generate a key pair $(\mathrm{SK}, \mathrm{PK}) \leftarrow \mathsf{Gen}(1^k)$. Send PK to Sender.
Chooser, $\mathsf{Query}_{\mathbf{PK}}(\sigma, n; \mathcal{Q})$**:**
For all $j \in [\alpha]$ do, for all $t \in [\lambda_j - 1]$ do:
Generate $r_{jt} \leftarrow_r \mathcal{R}$.
Set $\beta_{jt} \leftarrow \mathsf{Enc}_{\mathrm{PK}}^{s+(j-1)\xi}(\mathsf{Eq}(\sigma_j, t); r_{jt}) \in \mathcal{C}_{s+(j-1)\xi}$.
Send $\mathsf{Query}_{\mathrm{PK}}(\sigma, n; (r_{jt})_{j\in[\alpha], t\in[\lambda_j - 1]}) := (\beta_{jt})_{j\in[\alpha], t\in[\lambda_j-1]}$ to Sender.
Sender, $\mathsf{Transfer}_{\mathbf{PK}}(\mu, \mathsf{msg}_1; \mathcal{T})$**:**
For all $j \in [\alpha]$ do:
Set $\beta_{j,\lambda_j} \leftarrow \mathsf{Enc}_{\mathrm{PK}}^{s+(j-1)\xi}(1; 0) \cdot \prod_{t=1}^{\lambda_j - 1} \beta_{jt}^{-1} \mod \sharp\mathcal{C}_{s+(j-1)\xi}$.
Assume that $w_0 = \mu$ as an array.
For all $j \in \{1, \dots, \alpha - 1\}$ do:
For all $i_{j+1} \in [\lambda_{j+1}], \dots, i_\alpha \in [\lambda_\alpha]$ do:
Set $w_j(i_{j+1}, \dots, i_\alpha) \leftarrow \prod_{t\in[\lambda_j]} \beta_{jt}^{w_{j-1}(t, i_{j+1}, \dots, i_\alpha)} \mod \sharp\mathcal{C}_{s+(j-1)\xi}$.
Send $w_\alpha \leftarrow \prod_{t\in[\lambda_\alpha]} \beta_{\alpha t}^{w_{\alpha-1}(t)} \mod \sharp\mathcal{C}_{s+(\alpha-1)\xi}$
Send $\mathsf{Transfer}_{\mathrm{PK}}(\mu, \mathsf{msg}_1; \emptyset) := w_\alpha$ to Chooser.
Chooser $\mathsf{Recover}_{\mathbf{SK}}(\sigma, \mathsf{msg}_1, w'_\alpha)$**:**
For $j = \alpha$ downto 1 do:
Set $w'_{j-1} = \mathsf{Dec}_{\mathrm{SK}}^{s+(j-1)\xi}(w'_j)$.
Output w'_0.

Protocol 1: Protocol $\mathsf{LFCPIR}_d^n(\alpha)$, for fixed Π and fixed s

Protocol 1 depicts the new $\mathsf{LFCPIR}_d^n(\alpha)$ protocol that has parameters, optimised for large values of d. Note that here $\mathcal{Q} = \mathcal{R}^{\sum_{j\in[\alpha]}(\lambda_j - 1)}$ and $\mathcal{T} = \emptyset$. It is surprising that such a seemingly simple modification results in an important asymptotic improvement, stated by the next theorem.

Theorem 1. *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a length-flexible additively homomorphic cryptosystem that it is (ε, t)-secure in the sense of IND-LFCPA. Assume that $\mathcal{M}_{s+1} < d \le \mathcal{M}_s$ for some fixed $s \ge 1$, that Chooser has private input σ and Sender has private input $\mu = (\mu[1], \dots, \mu[n])$. Assume $\lambda_j = n^{1/\alpha}$ for all $j \in [1, \alpha]$.*

1. *For every $\alpha \in [n]$, there exists an (ε, t')-chooser-private and correct CPIR_d^n protocol $\mathsf{LFCPIR}_d^n(\alpha)$ with the chooser-side and the sender-side communication being respectively $\alpha(s + (\alpha + 1)\xi/2)(n^{1/\alpha} - 1) \cdot k$ and $(s + \alpha\xi) \cdot k$ bits. Here, $t' = t - t_{\mathsf{Cho}} - O(\alpha \cdot (sk)^{1+o(1)})$, where t_{Cho} is the time to execute the honest Chooser.*
2. *There exists an (ε, t')-chooser-private and correct CPIR_d^n protocol $\mathsf{LFCPIR}_d^n(\log n)$ with communication $(\xi/2 \cdot \log^2 n + (s + 3\xi/2) \cdot \log n + s) \cdot k = \Theta(\log^2 n \cdot k + \log n \cdot \log d)$. In this case, Chooser's workload is $t_{\mathsf{Cho}} = \Theta((s + \log n)k)^{2+o(1)})$ and Sender's workload is $t_{\mathsf{Sen}} = \Theta(n) \cdot (sk)^{2+o(1)}$. Therefore, $t' = t' - t_{\mathsf{Cho}} - O(\log n \cdot (sk)^{1+o(1)}) = t' - \Theta(((s + \log n)k)^{2+o(1)}))$.*

Proof. We will only prove the first item; the second one follows straightforwardly.

Correctness: clear, since $w'_{\alpha-1} = w_{\alpha-1}(\sigma_\alpha)$, $w'_{\alpha-2} = w_{\alpha-2}(\sigma_{\alpha-1}, \sigma_\alpha)$, $\dots$, $w'_{i-1} = w_{i-1}(\sigma_i, \dots, \sigma_\alpha), \dots$, and $w_0 = \mu(\sigma_1, \dots, \sigma_\alpha)$.

Communication: The chooser-side communication is $\sum_{j=1}^{\alpha} \sum_{t=1}^{\lambda_j - 1} (s + j\xi)k = \sum_{j=1}^{\alpha}(s + j\xi) \cdot (n^{1/\alpha} - 1) \cdot k = \alpha \left(s + \frac{(\alpha+1)\xi}{2}\right)(n^{1/\alpha} - 1) \cdot k$. We get the asymptotically optimal case in n if $\alpha = a \cdot \log n$ for some constant a, then the communication is $((-1/2 + 2^{1/a-1})a^2\xi \cdot \log^2 n + ((2^{1/a} - 1)as + (2^{1/a-1} + 1/2)a\xi) \cdot \log n + s) \cdot k$. This is minimised in $s \log n$ if $a = 1$, then the communication is $(\xi/2 \cdot \log^2 n + (s + 3\xi/2) \cdot \log n + s) \cdot k = \Theta(\log^2 n \cdot k + \log n \cdot \log d)$.

Computation (in the case (b)): Chooser must do $\lambda_j - 1$ encryptions $\mathsf{Enc}_{\mathrm{PK}}^{s+j-1}$ for any $j \in [n]$. Sender's work is dominated by $\left(\prod_{i=j+1}^{\alpha} \lambda_i\right)$ λ_j-term multi-exponentiations modulo M^{s+j} for every

$j \in [2, \alpha]$. Assume that a k-bit exponentiation can be done in time k^a for some a. Then, Sender's workload is dominated by $\sum_{i=1}^{\alpha-1} \lambda_i^{\alpha-i} \cdot (s+i)^a k^a$ bit-operations.

In the case (b), $\alpha = \log n$ and thus $\lambda_i = 2$ for every i. Then, Sender's workload is dominated by $\sum_{i=1}^{\alpha-1} 2^{\alpha-i} \cdot (s+i)^a k^a$ bit-operations. Asymptotically in n, this is equal to $\Theta(n) \cdot (sk)^a$. Fast exponentiation algorithms result in Sender's time $\Theta(n) \cdot (sk)^{2+o(1)}$. Therefore, Sender's computation cost in $\mathsf{LFCPIR}_d^n(\log n)$ is comparable to that of the 1-out-of-n oblivious transfer protocols from [NP01,AIR01] that have linear total communication. Chooser's work is even smaller, $\sum_{j=1}^{\alpha} \lambda_j k_{s+j-1}^a = 2 = \sum_{j=1}^{\log n-1} (s+j-1)^a k^a = \Theta(\sum_{j=1}^{\log n-1} (s^a + j^a) k^a) = \Theta((s+\log n)^a k^a = \Theta(((s+\log n)k)^{2+o(1)})$ bit-operations, when using asymptotically fast exponentiation algorithms.

Chooser-Privacy: Assume that some machine A that works in time t breaks the chooser-privacy of $\mathsf{LFCPIR}_d^n(\alpha)$ with probability ε. More precisely, a key pair (SK, PK) $\leftarrow \mathsf{Gen}(1^k)$ is generated. Given, PK, A generates $(\sigma_0, \sigma_1, \mu)$ and handles (σ_0, σ_1) to Chooser. Chooser picks a random bit $\overline{b}$ and sends the first message $\mathsf{Query}_{\mathsf{PK}}(\sigma_{\overline{b}}, n; r) = \{\beta_{jt}\}$ of the $\mathsf{LFCPIR}_d^n(\alpha)$ protocol, where r is randomly chosen from $\mathcal{Q}$, to A. A outputs a guess $\overline{b}'$, such that $|\Pr[\overline{b} = \overline{b}'] - \frac{1}{2}| \geq \varepsilon$.

Next, we construct a machine M that uses A as an oracle to break the IND-LFCPA security of Π with probability $\mathsf{Adv}_{\Pi,k}^{\mathsf{indcpa}}(M^A) = \varepsilon$. That is, given a random key pair (SK, PK), M comes up with a message pair (m_0, m_1) and an s, such that after seeing $\mathsf{Enc}_{\mathsf{PK}}^s(m_b; r)$ for a random $b \leftarrow \{0,1\}$ and $r \leftarrow \mathcal{R}$, he outputs a b', such that $|\Pr[b = b'] - \frac{1}{2}| \geq \varepsilon$.

M does the next: Let Chooser to generate (PK, SK), obtain PK and forward it to A. Obtain (σ_0, σ_1) from A. Assume that σ_0 and σ_1 differ in the coordinate set $\mathcal{J}$. This means that $\mathsf{Eq}(\sigma_{0j}, t) \neq \mathsf{Eq}(\sigma_{1j}, t)$ only if $j \in \mathcal{J}$ and $t \in \{\sigma_{0j}, \sigma_{1j}\}$.

M sets $m_0 \leftarrow 0$, $m_1 \leftarrow 1$ and asks for a challenge for coordinate set $\mathcal{J}$. That is, 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 $(\beta_{jt})_{jt}$ exactly as in Protocol 1, except that when $j \in \mathcal{J}$, he sets $\beta_{j,\sigma_{0j}} \leftarrow \mathsf{Enc}_{\mathsf{PK}}^{s+(j-1)\xi}(\mathsf{Eq}(\sigma_{0j}, t); \mathcal{R}) \cdot c_j$ and $\beta_{j,\sigma_{1j}} \leftarrow \mathsf{Enc}_{\mathsf{PK}}^{s+(j-1)\xi}(\mathsf{Eq}(\sigma_{1j}, t); \mathcal{R}) \cdot c_j$. Therefore, $(\beta_{jt})_{jt} = \mathsf{Query}(\sigma_b, n; \mathcal{Q})$. M sends $\{\beta_{jt}\}$ to A and obtains her guess $\overline{b}'$. M returns $b' = \overline{b}'$. Clearly, $b = b'$ if A guessed correctly. Therefore, M has success probability ε. M's time is equal to $t + t_{\mathsf{Cho}} + O(\alpha \cdot (sk)^{1+o(1)})$. □

Note that computation-efficiency is an important property of the $\mathsf{LFCPIR}_d^n(\alpha)$ protocol since otherwise in some applications one would prefer a protocol with linear communication but with a smaller computational complexity.

4.2 Optimisations

Optimisation for long documents. For large s's, $\mathsf{LFCPIR}_d^n(\alpha)$ gains even more on the competitors than for small s's. This is since while in most of the other CPIR's, the communication depends linearly on $\log d$—i.e., is equal to $\Theta(g(n,k) \log d)$ for some function g, in our case the dependency is, in a certain sense, sub-linear. Really, the communication of $\mathsf{LFCPIR}_d^n(\alpha)$ is $\Theta(\log^2 n \cdot k + \log n \cdot \log d)$. In particular, this means that there is left some space for trade-offs between communication and computation. Recall that Server's computational workload was $\Theta(n) \cdot (sk)^{2+o(1)}$. In the case of large $\log d \gg k$, one can say execute $\lfloor \log d/k \rfloor$ $\mathsf{LFCPIR}_d^n(\alpha)$'s in parallel, to retrieve elements from $\mathbb{Z}_d$ with communication $\Theta(\log^2 n \cdot k + \log n \cdot \log d + \log^2 d/k)$ and with Server's computation $\Theta(n \log d) k^{1+o(1)}$.

Optimisation for short documents. For short documents, we can do better. First, we can use the same protocol but with $\alpha = \alpha_0 \cdot \log n$ for $\alpha_0 = \frac{\ln 2}{W(-2e^{-2})+2} \approx 0.4349501937$; this gives the minimal communication $(0.3709430349 \cdot \xi \log^2 n + 1.705680513 \cdot s \cdot \log n + 1.287790450 \cdot \xi \cdot \log n + s) \cdot k = \Theta(\log^2 n \cdot k + \log n \cdot \log d)$ for small values of s.

We can do even better by redefining the values of λ_i as follows.

Corollary 1. *Fix* $\alpha \in [\log n]$. *Define* $\lambda_i := \left(\frac{(s+\alpha)!}{s!}\right)^{1/\alpha} \cdot \frac{1}{s+i} \cdot n^{1/\alpha}$. *There exists a* CPIR_d^n $\mathsf{LFCPIRHR}_d^n(\alpha)$ *with otherwise the same properties as* $\mathsf{LFCPIR}_d^n(\alpha)$ *but with the chooser-side and the sender-side communication being respectively* $((s+\alpha)!/s!)^{1/\alpha} \cdot \alpha \cdot (n^{1/\alpha} - 1) \cdot k$ *and* $(s+\alpha) \cdot k$ *bits,* *where* $t' = t - \Theta(((s+\log n)k)^{2+o(1)})$. *Fix* $\alpha_0 = \frac{\ln 2}{W(-2e^{-2})+2} \approx 0.4349501937$. *There exists a* CPIR_d^n *protocol* $\mathsf{LFCPIRHR}_d^n(\alpha_0 \cdot \log n)$ *with total communication* $(0.2729246327 \cdot \log n + (0.6274847945 \cdot s + 0.3137423972) \cdot \log\log n + O(1)) \log n \cdot k = \Theta(\log^2 n \cdot k + \log n \cdot \log\log n \cdot \log d)$.

This choice of λ_j results in the minimal value of $\sum_{j=1}^{\alpha}(\lambda_j - 1)(s+j) = \sum_{j=1}^{\alpha}\lambda_j(s+j) - \alpha(s + \frac{\alpha+1}{2})$ under the constraint that $\prod_{j=1}^{\alpha}\lambda_j = n$. (In practice, we must round λ_i-s to the nearest integers. For the simplicity of exposition, we will not explicitly mention such issues anymore.)

Really, Sender returns one element of length $(s+\alpha)k$. Chooser-side communication (the total length of ciphertexts β_{jt}) is $\sum_{j=1}^{\alpha}\sum_{t=1}^{\lambda_j - 1}(s+j)k = \sum_{j=1}^{\alpha}(\lambda_j - 1)(s+j)k = \sum_{j=1}^{\alpha}\sqrt[\alpha]{\frac{(s+\alpha)!}{s!}} \cdot (n^{1/\alpha} - 1)k = \sqrt[\alpha]{\frac{(s+\alpha)!}{s!}} \cdot \alpha \cdot (n^{1/\alpha} - 1)k$.

By Stirling's formula,

$$\begin{aligned}\left(\frac{(s+\alpha)!}{s!}\right)^{1/\alpha} &\approx \sqrt[\alpha]{\frac{\sqrt{2\pi(s+\alpha)}}{\sqrt{2\pi s}} \cdot \frac{(s+\alpha)^{s+\alpha}}{e^{s+\alpha}} \cdot \frac{e^s}{s^s}} \\ &= (s+\alpha)^{(2s+2\alpha+1)/(2\alpha)} s^{-(2s+1)/(2\alpha)} e^{-1} \\ &= \frac{\alpha}{e} + \frac{2s+1}{2e}\ln\alpha + O(1) \enspace .\end{aligned}$$

Therefore, chooser-side communication is $(\frac{\alpha}{e} + \frac{2s+1}{2e}\ln\alpha + O(1))\alpha(n^{1/\alpha - 1} - 1)k$. Ignoring the $O(1)$ term, we get that the asymptotically optimal case happens when $\alpha = \alpha_0 \cdot \log n$; in this case, the total communication is equal to $(0.2729246327 \cdot \log^2 n + 0.6274847945 \cdot s + 0.3137423972) \cdot \log n \cdot \log\log n + O(\log n)) \cdot k$.

If $s = 1$, $\mathsf{LFCPIRHR}_d^n(\alpha)$ is asymptotically approximately 1.347915861 times more communication-efficient than $\mathsf{LFCPIR}_d^n(\alpha)$. However, as s grows, $\mathsf{LFCPIR}_d^n(\alpha)$ becomes more and more efficient.

Optimisation by compressing. If $z := \lfloor sk/\log d \rfloor > 1$, one can use the next optimisation trick. Instead of $\mathsf{LFCPIR}_d^n(\alpha)$ with the query σ and the database $\mu = (\mu[1], \dots, \mu[n])$, execute $\mathsf{LFCPIR}_d^n(\alpha')$ (with suitably chosen α') with the query $\sigma' := \lfloor \sigma/z \rfloor$ and the database $\mu' = (\mu'[1], \dots, \mu'[\lfloor n/z \rfloor])$, where $\mu'[j]$ is the concatenation of z different consequent elements $\mu[\lceil j/z \rceil], \dots, \mu[\lceil j/z \rceil + z - 1]$ from the database μ. Thus, asymptotically, fixing $\alpha = \log(n/z)/2$, one can construct a CPIR_d^n with total communication $(0.2729246327 \cdot \log^2(n\frac{\log d}{sk}) + 0.4349393158 \cdot s \cdot \log(n\frac{\log d}{sk}) \cdot \log\log(n\frac{\log d}{sk}) + O(1)) \cdot k$. For example, if the database contains binary data, $\log d = 2$, this modified CPIR protocol has communication $\Theta((\log n - \log k)^2 + (\log n - \log k) \cdot \log(\log n - \log k)) \cdot k$.

While this optimisation does not result in an asymptotic improvement, the improvement can be quite important in practice. For example, when $n = k = 1024$, $d = 2$, the optimised version is 100 times more communication-efficient than the unoptimised version. The same optimisation was proposed by Chang [Cha04] in the case of Stern's CPIR_d^n, and it can also be applied to other protocols, e.g., to the AIR_d^n CPIR protocol from [AIR01].

Communication-balancing by repeating. Chang [Cha04] suggested another modification of Stern's CPIR_d^n that can be easily adapted to $\mathsf{LFCPIR}_d^n(\alpha)$. It increases the total communication, but helps one to trade off the chooser-side and the sender-side communication. Namely, divide Sender's database into n' equal parts. Execute $\mathsf{LFCPIR}_d^{n/n'}(\alpha')$, for suitable α', with query $\sigma' = \sigma \mod n'$ and every separate sub-database. The total chooser-side communication—assuming for the sake of simplicity that we have the hypercube presentation—is $\alpha' \cdot (s + \frac{\alpha'+1}{2})(n'^{1/\alpha'} - 1)k$, while the sender-side communication is $n/n' \cdot (s + \alpha') \cdot k$ bits.

4.3 Oblivious Transfer with Log-Squared Communication

Next, we discuss some techniques to transform the $\mathsf{LFCPIR}_d^n(\alpha)$ 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 chooser is semi-honest. For this it is sufficient that Sender masks all intermediate values

w_j by multiplying them with a random encryption of 0; this means that Sender's algorithm $\mathsf{Transfer}$ will not be deterministic anymore. We will next assume that Protocol 1 has been modified like that.

Using the Naor-Pinkas transformation. When using the Naor-Pinkas transformation from [NP99] on top of the $\mathsf{LFCPIR}^n_d(\alpha)$ protocol, we obtain straightforwardly a 1-out-of-n computationally chooser-private and computationally server-private oblivious transfer protocol $\mathsf{LFNPOT}^n_d(\alpha)$. If $\alpha = \log n$, the resulting oblivious transfer protocol $\mathsf{LFNPOT}^n_d(\log n)$ has communication $\Theta(\log^2 n \cdot k + \log n \cdot \log d)$. The $\mathsf{LFNPOT}^n_d(\alpha)$ protocol is secure, given that $\mathsf{LFCPIR}^n_d(\alpha)$ is computationally chooser-private (i.e., when the underlying length-flexible cryptosystem is IND-LFCPA secure) and when the used function F is pseudo-random.

Note that when doing that on top of Prot. 1, Sender can XOR the elements $\mu(t, i_2, \ldots, i_\alpha)$ with $\bigoplus_{j=1}^{\lambda} F_{K_j^{\sigma_j}}(\sigma)$; i.e., this does not need extra storage. This can be made computationally efficient by traversing the elements of the database by using a Gray code.

Construction based on zero-knowledge proofs. An alternative OT^n_d protocol can be constructed, based on zero-knowledge proofs of knowledge. In this case, Chooser must accompany PK also with a zero-knowledge proof that it has been correctly formed. First, Sender masks all values $w_j(i_{j+1}, \ldots, i_\alpha)$ by multiplying them with $\mathsf{Enc}^{s+j-1}_{\mathsf{PK}}(0; \mathcal{R})$. Second, Chooser must accompany the first message of the $\mathsf{LFCPIR}^n_d(\alpha)$ protocol with a standard honest-verifier zero-knowledge proof of knowledge (HVZK POK) that he chose the inputs correctly. This means (in the case of hypercube) $\alpha n^{1/\alpha}$ HVZK POK-s that an encrypted value is either 0 or 1, and $n^{1/\alpha}$ HVZK POK-s that the product of α ciphertexts decrypts to 1. Again, some of these HVZK POK-s are done with a larger s, so that the total overhead of the HVZK POK-s is $\Theta(\log^2 n)k$. This does not increase the asymptotic communication complexity of the protocol. Therefore, we can construct a two-round (i.e., four-message) OT^n_d protocol with total communication $\Theta(\log^2 n)k$ that can be made one-round in the random-oracle model by using the Fiat-Shamir heuristic [FS86]. In the latter case, the non-interactive HVZK POK-s result only in logarithmic overhead.

Next, we give a detailed description of the HVZK POK that $\mathsf{Dec}^s_{\mathsf{SK}}(c) \in \{0, 1\}$. For this, one first computes $c_1 = c/\mathsf{Enc}^s_{\mathsf{PK}}(0; 0)$ and $c_2/\mathsf{Enc}^s_{\mathsf{PK}}(1; 0)$ and then proves that either $\mathsf{Dec}^s_{\mathsf{SK}}(c_1) = 0$ or $\mathsf{Dec}^s_{\mathsf{SK}}(c_2) = 0$, as follows. Let Π be an IND-LFCPA secure additively homomorphic cryptosystem and let $\tau \approx 80$. Assume that $c_1 = \mathsf{Enc}^s_{\mathsf{PK}}(0; r)$. Let c_2 be another ciphertext. The next protocol for proving that either $\mathsf{Dec}^s_{\mathsf{SK}}(c_1) = 0$ or $\mathsf{Dec}^s_{\mathsf{SK}}(c_2) = 0$ was proposed in [DJ01]:

1. Chooser chooses a random $r_1 \leftarrow \mathcal{R}$ and sets $a_1 \leftarrow \mathsf{Enc}^s_{\mathsf{PK}}(0; r_1)$. He generates a view (a_2, e_2, z_2) by setting $e_2 \leftarrow \{0,1\}^\tau$, $z_2 \leftarrow \mathcal{R}$ and $a_2 \leftarrow \mathsf{Enc}^s_{\mathsf{PK}}(0; z_2) \cdot c_2^{-e_2} \mod M^{s+1}$. He sends (a_1, a_2) to Sender.
2. Sender chooses a random τ-bit number e, and sends e to Chooser.
3. Chooser computes $e_1 \leftarrow e - e_2 \mod 2^\tau$ and $z_1 \leftarrow r_1 r^{e_1} \mod M$. He sends (e_1, z_1, z_2) to Sender.
4. Sender computes $e_2 \leftarrow e - e_1 \mod 2^\tau$ and checks that $\mathsf{Enc}^s_{\mathsf{PK}}(0; z_1) = a_1 c_1^{e_1} \mod M^{s+1}$, $\mathsf{Enc}^s_{\mathsf{PK}}(0; z_2) = a_2 c_2^{e_2} \mod M^{s+1}$, and that $u_1, u_2, a_1, a_2, z_1, z_2$ are relatively prime to M. He accepts iff all verifications succeed.

The communication of this HVZK POK is $2 \log \sharp\mathcal{C}_s + 2\tau + 2 \log \sharp R_s = 2(s + \xi)k + 2\tau + 2k = 2(s + \xi + 1) \cdot k + 2\tau$ bits. When using the Fiat-Shamir, only (e_1, z_1, e_2, z_2) must be transferred, or $2 \log \sharp R_s + 2\tau = 2k + 2\tau$ bits.

Therefore, the non-interactive zero-knowledge proof that all $\prod_{i=1}^{\alpha}(\lambda_i - 1)$ ciphertexts decrypt either to 0 or 1, $(e_1, e_2, z_{11}, \ldots, z_{\alpha,\lambda_\alpha - 1})$, takes $2k \sum_{i=1}^{\alpha}(\lambda_i - 1) + 2\tau$ bits. Assuming $\alpha = \log n$ and $\lambda_i = 2$, this is equal to $2 \log n \cdot k + 2\tau$ bits.

The α non-interactive zero-knowledge proofs that some product of ciphertexts decrypt to 1 adds $\alpha k + \tau$ bits, but these proofs can be omitted if $\alpha = 2$ and $\lambda_i = 2$.

Theorem 2. *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be the* DJ01 *cryptosystem. Assume that the DCRP is hard. For every $\alpha \geq 1$, there exists an 1-out-of-n oblivious transfer protocol $\mathsf{LFZKOT}^n_d(\alpha)$ with the communication $(\alpha(s + \frac{(\alpha+1)\xi}{2})(n^{1/\alpha} - 1) + s + \alpha\xi + 2\sum_{i=1}^{\alpha}(\lambda_i - 1) + \alpha)k + 3\tau$ bits. In particular, there exists an OT^n_d protocol $\mathsf{LFZKOT}^n_d(\log n)$ with total communication $(\xi/2 \cdot \log^2 n + (s + 3\xi/2 + 2) \cdot \log n + s) \cdot k + 2\tau$ bits. The $\mathsf{LFZKOT}^n_d(\alpha)$ protocol is secure in the random oracle model, assuming that Π is IND-CPA secure.*

The proof of this theorem is standard.
Construction based on additive disclosure of secrets. Recently, Laur and Lipmaa [LL04] proposed another transformation, called *additive disclosure of secrets* for turning $\mathsf{HomCPIR}_d^n(\alpha)$ into an oblivious transfer protocol. The same methodology can be used in the case of the $\mathsf{LFCPIR}_d^n(\alpha)$ protocol, although the resulting protocol is a bit more involved. The short idea is as follows. (Note that also in this case, Chooser must prove in zero-knowledge that PK is correctly formed, but this can be done once, in the initialisation phase.) Every input β_{jt}, sent by Chooser, is an encryption of 0 or 1. For every valid (j, t), Chooser prepares a two-element database $\mathsf{Mask}_{j,t} = (\mathsf{mask}_{j,t}, \mathsf{mask}_{j,t})$, where $\mathsf{mask}_{j,t}$ is a sufficiently large random number. For every Chooser's input β_{jt}, Chooser and Sender then execute a new 1-out-of-2 oblivious transfer protocol, proposed in [LL04], with $\mathsf{msg}_{jt,1} := \beta_{jt}$. Additionally, they execute the same oblivious transfer protocol with $\mathsf{msg}_{j0,1} := \prod_{i=1}^{\alpha-1} \beta_{jt}$. Sender, instead of sending the result $\mathsf{msg}_{jt,2}$ to Chooser, stores it. (The latter can be again omitted if $\alpha = \log n$ and $\lambda_i = 2$.) The oblivious transfer protocol has the property that if the query is either 0 or 1 then the stored value encrypts $\mathsf{mask}_{j,t}$, otherwise it encrypts a sufficiently large random value. After that, Sender executes the code for the $\mathsf{LFCPIR}_d^n(\alpha)$ protocol, but masks all the internal values $w_j(i_{j+1}, \dots, i_\alpha)$ by multiplying them with $\prod_{t\in[0,\lambda_j-1]} \mathsf{msg}_{jt,2}$. The resulting oblivious transfer protocol has *one round*, is *information-theoretically server-private* if Π is IND-LFCPA secure, i.e., *in the standard model*.

5 Two Hypothetical Logarithmic CPIR_d^n-s

The constructed CPIR_d^n $\mathsf{LFCPIR}_d^n(\alpha)$ has total communication $\Theta(\log^2 n \cdot k + \log n \cdot \log d)$. However, the communication complexity of the non-private information retrieval is not more than $\log n + \log d$ (Chooser sends a $(\log n)$-bit index σ and Sender responds with a $\log d$-bit value μ_σ).

Next, we modify $\mathsf{LFCPIR}_d^n(\alpha)$, so as to have near-optimal communication $\Theta(\log n \cdot (k + \log d))$ under some feasible additional conjectures on the used homomorphic cryptosystem. We do not know whether such cryptosystems exist but we would like to think that they do. The common motivation behind both modifications is based on the next observation: $\mathsf{LFCPIR}_d^n(\alpha)$ has log-squared—and not logarithmic—communication mostly because the length of the values β_{jt} depends (linearly) on j. Thus, one way to improve the communication-efficiency $\mathsf{LFCPIR}_d^n(\alpha)$ to logarithmic is to make $|\beta_{jt}|$ independent of j. We propose two different modifications to the $\mathsf{LFCPIR}_d^n(\alpha)$ that overcome this obstacle, although under two different conjectures.

In the first case, we assume that $\beta_{jt} = \mathsf{Enc}_{\mathsf{PK}}^{s+j-1}(\mathsf{Eq}(\sigma_j, t); r_{jt})$ can be securely communicated to Sender by using ηsk bits, i.e., that $|\beta_{jt}|$ does not depend on j at all. This involves an hypothesis of the existence of a certain succinct cryptosystem. In the second solution, we assume that $\beta_{jt} = \mathsf{Enc}_{\mathsf{PK}}^{s}(\mathsf{Eq}(\sigma_j, t); r_{jt})$ is used as a multiplicand (and not as an exponent); this involves an alternative hypothesis of the existence of a ring-homomorphic cryptosystem. We conjecture that such cryptosystems exist—although we also show that the DJ01 cryptosystem does not have the conjectured properties. We hope that spelling out these two conjectures gives an extra motivation for studying cryptosystems with such properties.
First case. We now make the next conjecture.

Conjecture 1. There exists a length-flexible homomorphic cryptosystem $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ that is secure in the sense of IND-LFCPA, such that (a) $\mathcal{C}_s \subseteq \mathcal{M}_{s+1}$, (b) $\log \sharp\mathcal{C}_s \approx (s + 1)k$ and (c) for every s, z there exist efficient functions $\mathsf{Hide}_s : \mathcal{M}_s \times \mathcal{R} \to Z_s$ and $\mathsf{Expand}_{s,z} : Z_s \to \mathcal{C}_{s+z}$, such that $\mathsf{Expand}_{s,z}(\mathsf{Hide}_s(m; r)) = \mathsf{Enc}_{\mathsf{PK}}^{s+z}(m; r')$, for $m \in \{0, 1\}$ and for any $r \in \mathcal{R}$. Moreover, $\log \sharp Z_s = \log \sharp\mathcal{C}_s = (1 + o(1))s \log M$ for some M. We call such a cryptosystem *succinct*.

(Here, (b) can be softened to the requirement that $\log \sharp\mathcal{C}_s \approx (s + \gamma) \cdot k$ for some constant γ.) Based on a succinct cryptosystem Π, we can construct a new CPIR_d^n protocol, that looks almost exactly the same as Protocol 1, except that (a) β_{jt} is defined as $\beta_{jt} \leftarrow \mathsf{Hide}_s(\mathsf{Eq}(\sigma_j, t); r_{jt})$, and (b) when using $\beta_{i,t}$ in the i-th sub-round of computing the second message, Sender first applies the function Expand_i on it. As an example, he sets $w_\alpha \leftarrow \prod_{t\in[\lambda_\alpha]} \mathsf{Expand}_\alpha(\beta_{\alpha,t})^{w_{\alpha-1}(t)} \mod M^{s+\alpha}$. Thus, Chooser communicates every β_{jt} to Sender by using ηsk bits. Sender just sends back the value w_α.

Lemma 2. *Let Π be a (hypothetical) succinct cryptosystem. (a) For every $\alpha \in [n]$, there exists a* CPIR_d^n *protocol* $\mathsf{HypoCPIR1}_d^n(\alpha)$ *with chooser-side and sender-side communication being respectively* $(s+1)\alpha(n^{1/\alpha}-1)k$ *and* $(s+\alpha)k$ *bits. (a) There exists a* CPIR_d^n *protocol* $\mathsf{HypoCPIR1}_d^n(\log n)$ *with chooser-side and sender-side communication being respectively* $(s+1)\log n \cdot k$ *and* $(s+\log n)\cdot k$ *bits.*

Proof. Correctness: By Conjecture 1, $\mathsf{Expand}_j(\beta_{jt}) \mod M^{s+j} = \mathsf{Enc}_{\mathsf{PK}}^{s+j-1}(\mathsf{Eq}(\sigma_j,t);r_{jt})$. *Communication*: chooser-side communication (the total length of ciphertexts β_{jt}) is $\alpha\cdot(n^{1/\alpha}-1)(s+1)\cdot k$. *Privacy*: Sender sees only random encryptions of 0-s and 1-s, and thus Chooser's privacy is guaranteed by the IND-LFCPA security of Π. □

For $s>1$, the communication is minimised when $\alpha = \log n$. For $s=1$, the actual optimum is obtained when $\alpha = \frac{\ln 2}{1+W(-(2e)^{-1})}$. For $s=1$, the communication of $\mathsf{HypoCPIR1}_d^n(\log n)$ is equal to $(1+3\cdot\log n)\cdot k$ bits.

On the (im)possibility of succinct cryptosystems. Since Π is an IND-LFCPA secure homomorphic cryptosystem, $\mathsf{Enc}_{\mathsf{PK}}^s(m_1;r_1)\cdot\mathsf{Enc}_{\mathsf{PK}}^s(m_2;r_2)=\mathsf{Enc}_{\mathsf{PK}}^s(m_1+m_2;r_1\circ r_2)$. For all IND-LFCPA-secure homomorphic cryptosystems, $\cdot$ corresponds to the modular multiplication in $\mathcal{C}_s=\mathbb{Z}_{M_s}^*$ for some M_s. Thus, in such a case $\mathsf{Enc}_{\mathsf{PK}}^s(m;r)=g_s^m\psi(r,s) \mod M_s$ for some element $g_s\in\mathbb{Z}_{M_s}^*$ and for some function ψ. Moreover, when Π is length-flexible then it holds—at least for the currently known length-flexible cryptosystems Π [DJ01,DJ03]—that $\overline{M} := \sharp\mathcal{C}_{s+1}/\sharp\mathcal{C}_s$ is an integer, and that $\mathsf{Enc}_{\mathsf{PK}}^{s+1}(m;r) = g_{s+1}^m\psi(r,s+1) \mod M_{s+1}$, where $M_{s+1}=\overline{M}M_s$ for some integer $\overline{M}$.

Assume now that $\mathsf{Enc}_{\mathsf{PK}}^s(m;r)=a \mod M_s$ (e.g., $a=g_s^m\psi(r,s)$ without modular reduction), that is, $a=\mathsf{Enc}_{\mathsf{PK}}^s(m;r)+bM_s$ for some b. We want to establish for which functions h, $h(a) \mod M_{s+1}$ can be computed from $a \mod M_s$—that is, for which functions h, $h(a) \mod M_{s+1}$ does not depend on b.

When we assume that h does not involve modular reductions, we can easily find that h must be one of the next basic operations or their composite. First, $h_1(a)=a\overline{M}$. Then, $h_1(a)\equiv a\overline{M}\equiv\mathsf{Enc}_{\mathsf{PK}}^s(m;r)\overline{M} \mod M_{s+1}$, thus $h_1(a) \mod M_{s+1}$ can be computed from $a \mod M_s$. Second, $h_2(a)=a^{\overline{M}}$, since $h_2(a)\equiv\mathsf{Enc}_{\mathsf{PK}}^s(m;r)^{\overline{M}} \mod M_{s+1}$. Assuming $\mathsf{Enc}_{\mathsf{PK}}^{s+1}(m;r)=g_s^m\psi(r,s+1) \mod M_{s+1}$, we get $a^{\overline{M}}\equiv g_s^{\overline{M}m}\psi(m,s+1)^{\overline{M}} \mod M_{s+1}$.

No other function on $a \mod M_s$, a is an arbitrary integer, except the two cases above (h_1 and h_2) and their compositions with each other and with functions in $\mathbb{Z}_{M_{s+1}}$, can return a meaningful function $h(a) \mod M_{s+1}$: e.g., for $\overline{M}\nmid j$, $a^j\equiv(\mathsf{Enc}_{\mathsf{PK}}^s(m;r)+bM_s)^j\equiv(\mathsf{Enc}_{\mathsf{PK}}^s(m;r)^j+\binom{j}{2}\mathsf{Enc}_{\mathsf{PK}}^s(m;r)^{j-1}b+\dots) \mod M_{s+1}$ depends also on b—a value that is unknown to the potential Sender.

Thus, $\mathsf{HypoCPIR1}_d^n(\alpha)$ exists iff some composition of h_1 and h_2 can transform a valid ciphertext in $\mathcal{C}_s$ to a valid ciphertext of the same element in $\mathcal{C}_{s+1}$. Let us look at the case of the DJ01 cryptosystem.

Lemma 3. *Let $\Pi=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ be the* DJ01 *cryptosystem. Let $\alpha=\mathsf{Enc}_{PK}^s(m;r)$ for some $m\in\mathbb{Z}_{M^s}$ and $r\in\mathbb{Z}_n^*$. Then, $\mathsf{Enc}_{PK}^{s+1}(mM;r)=\alpha^M \mod M^{s+2}$ and thus $\mathsf{Enc}_{PK}^{s+\alpha}(mM^\alpha;r)=\alpha^{M^\alpha} \mod M^{s+\alpha+1}$ for any $\alpha>0$.*

Proof. First, $\alpha=(1+M)^m r^{M^s} \mod M^{s+1}$. Thus, $\alpha^M\equiv((1+M)^m r^{M^s} \mod M^{s+1})^M \mod M^{s+2}$. But $((1+M)^m \mod M^{s+1})^M\equiv\left(1+mM+\dots+\binom{m}{s}M^s\right)^M\equiv(1+M)^{mM} \mod M^{s+2}$ and $(r^{M^s} \mod M^{s+1})^M\equiv r^{M^{s+1}} \mod M^{s+2}$. Therefore, $\alpha^M\equiv(1+M)^{mM}r^{M^{s+1}} \mod M^{s+2}$. □

In this case, $\overline{M}=M$. Then, $\mathsf{Enc}_{\mathsf{PK}}^s(m;r)\cdot M \mod M^{s+2}$ is not a valid ciphertext and $\mathsf{Enc}_{\mathsf{PK}}^s(m;r)^M\equiv\mathsf{Enc}_{\mathsf{PK}}^{s+1}(mn;r) \mod M^{s+2}$. The latter value cannot be used to compute $\mathsf{Enc}_{\mathsf{PK}}^{s+1}(m;r)$ by somebody who does not know the factorisation of M, since M is not an invertible element of $\mathcal{M}_{s+1}$. Thus, the DJ01 cryptosystem is not succinct.

An hypothetical CPIR based on a ring-homomorphic cryptosystem. Assume that Π is a ring-homomorphic cryptosystem, that is, $\mathsf{Dec}_{\mathsf{SK}}^s(\mathsf{Enc}_{\mathsf{PK}}^s(m_1;r_1)\otimes\mathsf{Enc}_{\mathsf{PK}}^s(m_2;r_2))=m_1\cdot m_2$ and $\mathsf{Dec}_{\mathsf{SK}}^s(\mathsf{Enc}_{\mathsf{PK}}^s(m_1;r_1)\oplus\mathsf{Enc}_{\mathsf{PK}}^s(m_2;r_2))=m_1+m_2$ where $\mathcal{C}_s$ is a finite ring with operations $(\otimes,\oplus)$

and $\mathcal{M}_s$ is a finite ring with operations $(\cdot, +)$. In this case, we again modify Protocol 1 just by changing the definitions of β_{jt} and w_i. First, set $\beta_{jt} \leftarrow \mathsf{Enc}^s_{\mathsf{PK}}(\mathsf{Eq}(\sigma_j, t); r_{jt})$. Then, set, e.g.,

$$w_1(i_2, \dots, i_\alpha) \leftarrow \bigoplus_t \left(\mathsf{Enc}^s_{\mathsf{PK}}(\mu(t, i_2, \dots, i_\alpha); r'_{i_2,\dots,i_\alpha}) \otimes \beta_{1,t}\right) \quad ,$$

thus $\mathsf{Dec}^s_{\mathsf{SK}}(w_1(i_2, \dots, i_\alpha)) = \mu(\sigma_1, i_2, \dots, i_\alpha)$. Computation of w_i for $i > 1$ is analogous. The resulting hypothetical CPIR^n_d $\mathsf{HypoCPIR2}^n_d(\alpha)$ is slightly more efficient than $\mathsf{HypoCPIR1}^n_d(\alpha)$ since the sender only returns one ciphertext from $\mathcal{C}_s$ and not from $\mathcal{C}_{s+\alpha}$, as in the case of $\mathsf{HypoCPIR1}^n_d(\alpha)$.

Lemma 4. *Let Π be a (hypothetical) ring-homomorphic cryptosystem. (a) For every $\alpha \in [n]$, there exists a CPIR protocol $\mathsf{HypoCPIR2}^n_d(\alpha)$ with chooser-side and sender-side communication being respectively $\alpha(n^{1/\alpha} - 1)(s + 1)k$ and $(s + 1)k$ bits. (a) Let $\alpha = \log n$. There exists a CPIR protocol $\mathsf{HypoCPIR2}^n_d(\alpha)$ with the chooser-side and the sender-side communication being respectively $(s + 1) \cdot \log n \cdot k$ and $(s + 1) \cdot k$ bits.*

For $s = 1$, the chooser-side communication of $\mathsf{HypoCPIR2}^n_d(\log n)$ is $2 \cdot \log n \cdot k$ and the sender-side communication is $2 \cdot k$ bits. However, the Damgård-Jurik cryptosystem is not ring-homomorphic and moreover, no such cryptosystems are currently known. Our intuition is that it is "easier" to construct a succinct cryptosystem than to construct a ring-homomorphic cryptosystem. One of the reasons for this intuition is that it is well-known that a ring-homomorphic cryptosystems would enable to implement very efficiently many other tasks (including general multi-party computation), while the existence of a succinct cryptosystem has not so many direct corollaries.

6 Comparisons and Discussion

6.1 On Concrete Versus Polynomial Security

It is necessary to use concrete security (i.e., talking about adversaries, working in time t and breaking a primitive with probability ε) when one wants to be able to quantify the value of the used security parameter. There is a distinct reason why concrete security should be preferred over the polynomial security (i.e., talking about adversaries who work in time, polynomial in the size of the input, and break a primitive with probability, negligible in the size of the input) when one reasons about a CPIR^n_d protocol with polylogarithmic communication.

Namely, the input size of a CPIR^n_d protocol is $\log d \cdot n$. Assume, for a change, that no adversary is able to break the CPIR^n_d in time, polynomial in n, with a non-negligible probability in n. Assume also that the underlying hard problem, given inputs M of size k, can be broken in time $L_M[a, b]$. (In the case of $\mathsf{LFCPIR}^n_d(\alpha)$, when based on the DJ01 cryptosystem, $b = 1/3$.) Then, it is necessary that $L_M[a, b] = \omega(n^c)$ for every constant c, or $k^b \log^{1-b} k = \omega(\log n)$. Omitting the logarithmic factor, we get that $k = \Omega(\log^{1/b} n)$. Therefore, in our case, when basing $\mathsf{LFCPIR}^n_d(\alpha)$ on the DCRP, we must assume that $k = \Omega(\log^{3-o(1)} n)$ and thus the communication of the $\mathsf{LFCPIR}^n_d(\log n)$ becomes $\Theta(\log^{5-o(1)} n + \log n \cdot \log d)$.

However, such an analysis is controversial for several reasons. First, the size of the database could be anything from $n = 2$ to say $n = 2^{32}$. With such a large range of possible input lengths, we do not feel that a polynomial in n would capture correctly the security requirements one has in the real-life: after all, one wants to get assurance, that the CPIR cannot be broken—say—in time $t = 2^{90}$ with probability higher than 2^{-90}. This should hold even if $n = 2$ and more security is rarely needed even if say $n = 2^{32}$. If k would be, say, equal to $\log^3 n$), we would get $k = 1$ if $n = 2$ and $k = 2^{15}$ if $n = 2^{32}$, both values of k differ from what one actually wants from a CPIR in practice.

Second, a value that is polynomial in k is also polynomial in k^2 and $\sqrt{k}$ and therefore one can vary k's value and thus claim that the CPIR protocol has communication $\Theta(k^\varepsilon)$ for almost any value of ε. This makes it hard to estimate the precise communication of the protocol. Third, while importantly, polynomial security approach captures the hardness of breaking the underlying problem into the estimated communication, it is easier and more logical to define another security parameter, κ, corresponding to the desideratum that breaking the CPIR should be hard in time 2^κ, and then expressing the

Protocol	Chooser-side	Sender-side
Without privacy	$\lceil \log n \rceil$	$\lceil \log d \rceil$
$\mathsf{HomCPIR}_d^n(\alpha)$	$\eta\alpha(n^{1/\alpha}-1)\cdot sk$	$\eta^\alpha \cdot k$
$\mathsf{HomCPIR}_d^n(\log_\eta^{1/2} n)$	$\eta \log_\eta^{1/2} n \cdot (\eta^{\sqrt{\log_\eta n}}-1)\cdot k$	$\eta^{\log_\eta^{1/2} n}\cdot k$
CMS_d^n	$\kappa^{2f}+2\kappa^4$	$\kappa^4 \cdot \log d$
$\mathsf{LFCPIR}_d^n(\alpha)$	$\alpha(s+(\alpha+1)/2)(n^{1/\alpha}-1)\cdot k$	$(s+\alpha)\cdot k$
$\mathsf{LFCPIR}_d^n(\log n)$	$\xi/2\cdot\log^2 n\cdot k+((s+\xi/2)\cdot\log n)\cdot sk$	$(s+\log n)\cdot k$
$\mathsf{LFCPIRHR}_d^n(\alpha)$	$\left(\frac{(s+\alpha)!}{s!}\right)^{1/\alpha}\cdot\alpha\cdot(n^{1/\alpha}-1)k$	$(s+\alpha)\cdot k$
$\mathsf{LFCPIRHR}_d^n(\alpha_0\cdot\log n)$	$\approx 0.273\log^2 n\cdot k+0.435\cdot\log n\cdot\log\log n\cdot sk$	$(s+\alpha_0\cdot\log n)\cdot k$
$\mathsf{HypoCPIR1}_d^n(\alpha)$	$(s+1)\cdot\alpha(n^{1/\alpha}-1)\cdot k$	$(s+\alpha)\cdot k$
$\mathsf{HypoCPIR1}_d^n(\log n)$	$(s+1)\log n\cdot k$	$(s+\log n)\cdot k$
$\mathsf{HypoCPIR2}_d^n(\alpha)$	$(s+1)\cdot\alpha(n^{1/\alpha}-1)\cdot k$	$(s+1)\cdot k$
$\mathsf{HypoCPIR2}_d^n(\log n)$	$(s+1)\cdot\log n\cdot k$	$(s+1)\cdot k$

Table 1. Total communication of some CPIR protocols when retrieving $\lceil \log d\rceil < sk$ bits with a single query. For simplicity, we omit the ceiling signs, but it must be remembered that α is an integer. Here, $\alpha_0 = \frac{\ln 2}{W(-2e^{-2})+2} \approx 0.4349501937$ as defined in Thm. 1 and κ is a security parameter s.t. $\kappa \geq \log^2 n$.

communication in the terms of κ. In the case of the DCRP, based on the hypothesis that the best attack against it is the general umber field sieve, it would mean that $k\cdot(\ln k)^2 = \Omega(\frac{9(\ln 2)^2\kappa^3}{64}) = \Omega(\kappa^3)$ and thus $\mathsf{LFCPIR}_d^n(\alpha)$ has communication $\Theta(\log^2 n\cdot\kappa^{3-o(1)}+\log n\cdot\log d)$.

6.2 Communication

Fix $k = 1024$ and $s = 1$. Table 1 summaries the next discussion.

The difference between the communications of the linear Aiello-Ishai-Reingold CPIR AIR_d^n [AIR01] (with communication $2(n+1)k$), the polylogarithmic Cachin-Micali-Stadler CPIR [CMS99] (with overly optimistic setting $k = \min(80, \log^2 n)$ and $f = 4$; whether the CMS_d^n CPIR is actually secure in this setting is unknown), the super-polylogarithmic $\mathsf{HomCPIR}_d^n(\log^{1/2} n)$, and the new CPIR's, $\mathsf{LFCPIR}_d^n(\frac{\ln 2}{W(-2e^{-2})+2}\cdot\log n)$, $\mathsf{LFCPIRHR}_d^n(\frac{\ln 2}{W(-2e^{-2})+2}\cdot\log n)$—and some of their optimisations—and the hypothetical $\mathsf{HypoCPIR1}_d^n(\log n)$ is depicted by the Figure 1, Figure 2, Figure 3, Figure 4. To be completely fair, we have applied our basic optimisation idea (of not sending the λ_i-th element of dimension i to Sender since Sender can compute it himself) also to $\mathsf{HomCPIR}_d^n(\log^{1/2} n)$; this results in communication, provided in Table 1. Note that the hyper-rectangle-based $\mathsf{LFCPIR}_d^n(\frac{\ln 2}{W(-2e^{-2})+2}\cdot\log n)$ is more communication-efficient than $\mathsf{HomCPIR}_d^n(\log^{1/2} n)$ for all n-s.

Acknowledgements. We would like to thank Yan-Cheng Chang, Sven Laur and anonymous referees for useful comments. This work was partially supported by the Finnish Defence Forces Institute for Technological Research and by the Finnish Academy of Sciences.

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, 6–10 May 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, 13–15 July 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*,

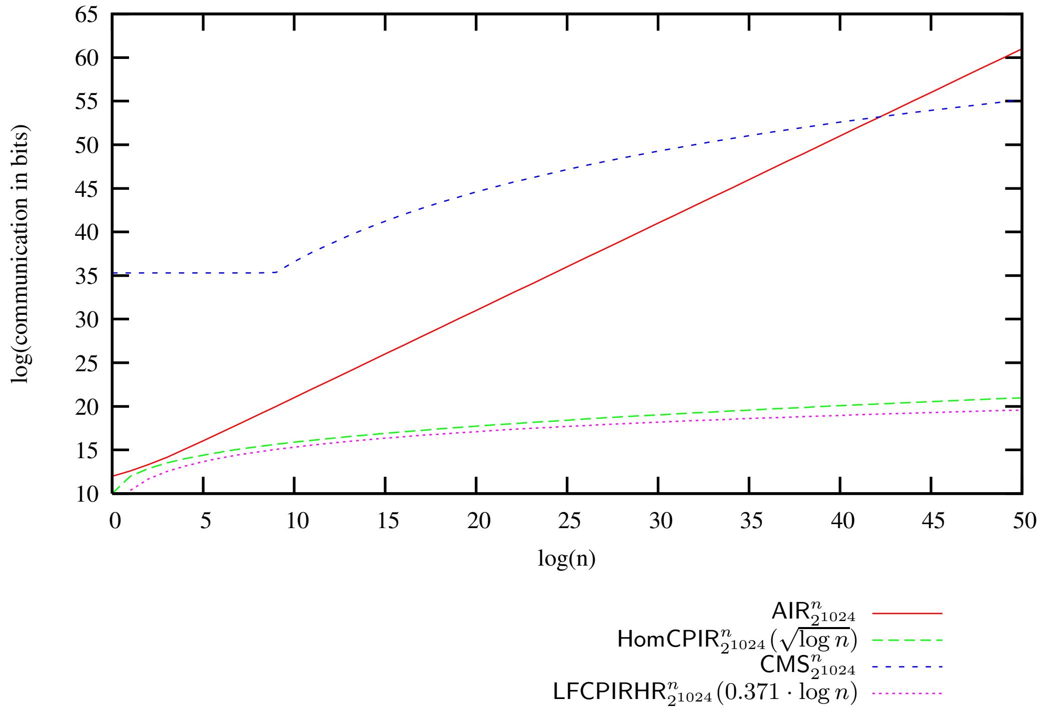

Fig. 1. Logarithm of communication of some of the previously known CPIR's on the logarithmic scale in n, assuming that $k = 1024$ and $\eta = 2$. (Except for the CMS^n_2 CPIR, where $k = 80$.)

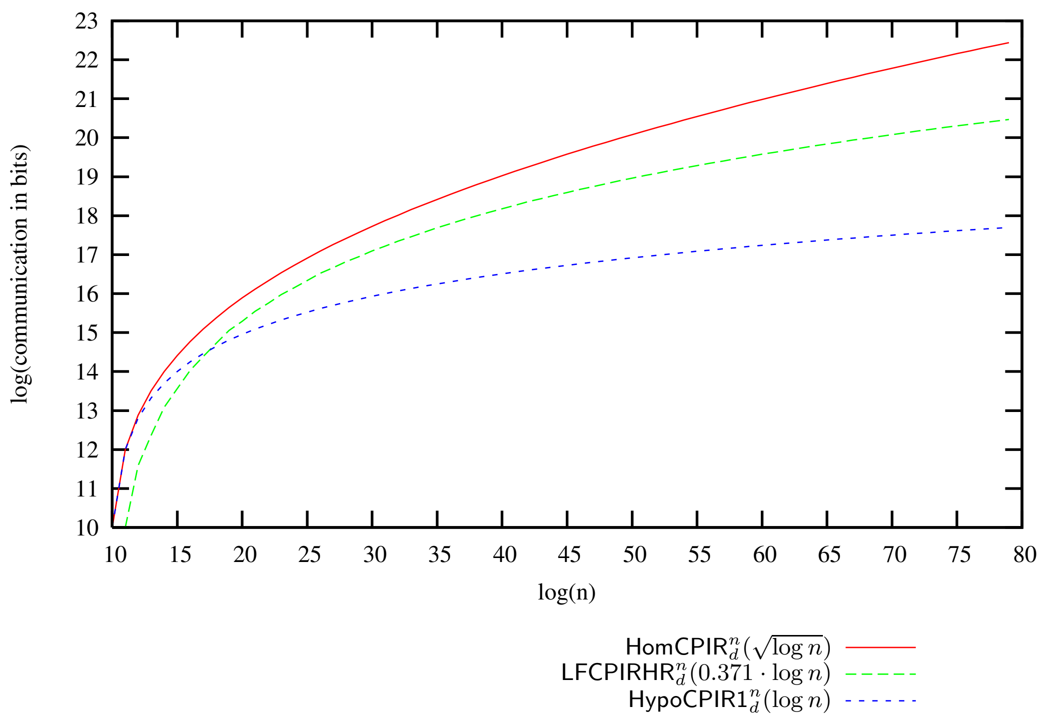

Fig. 2. Logarithm of communication of some of $\mathsf{HomCPIR}^n_d(\alpha)$ and some of the new CPIR's on the logarithmic scale in n, assuming that $k = 1024$, $\eta = 2$ and $d = 2$

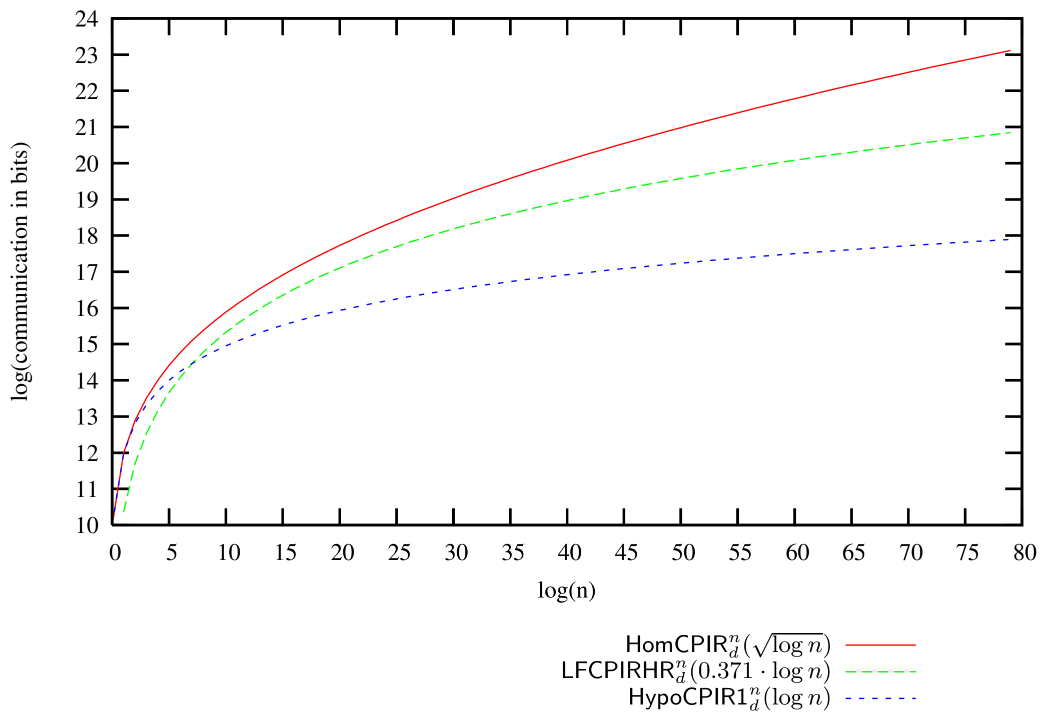

Fig. 3. Logarithm of communication of some of $\mathsf{HomCPIR}^n_d(\alpha)$ and some of the new CPIR's on the logarithmic scale in n, assuming that $k = 1024$, $\eta = 2$ and $d = 2^{1024}$

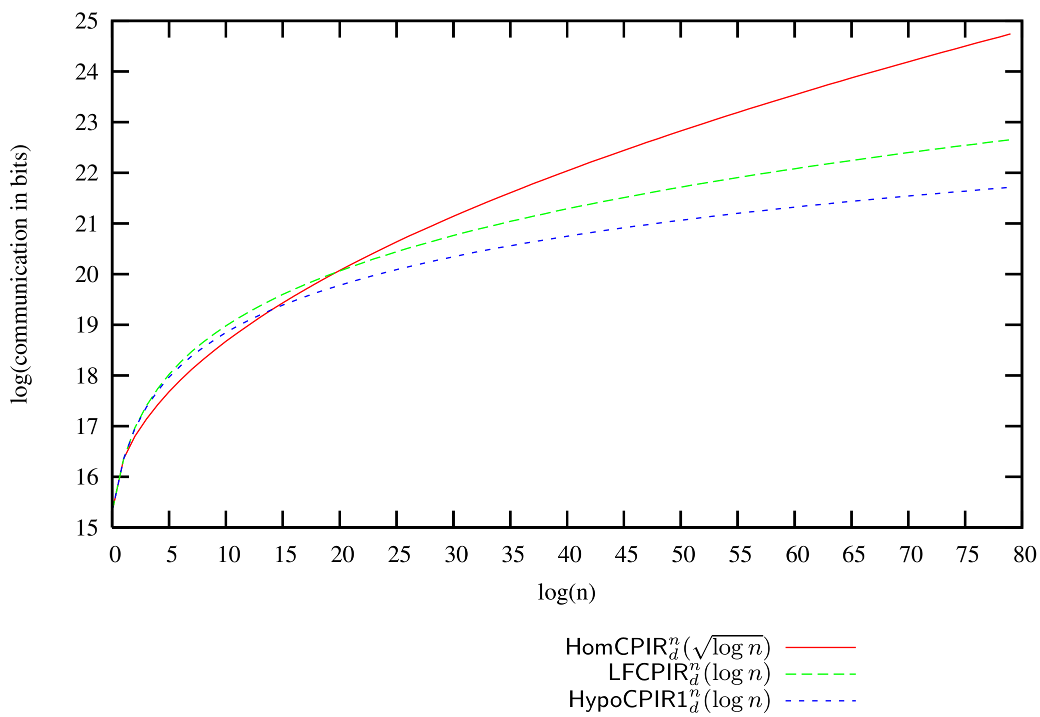

Fig. 4. Logarithm of communication of some of $\mathsf{HomCPIR}^n_d(\alpha)$ and some of the new CPIR's on the logarithmic scale in n, assuming that $k = 1024$, $\eta = 2$ and $d = 2^{40960}$

volume 1592 of *Lecture Notes in Computer Science*, pages 402–414, Prague, Czech Republic, 2–6 May 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, 13–15 February 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.

[FS86] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology—CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, California, USA, 11–15 August 1986. Springer-Verlag, 1987.

[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, 20–22 October 1997. IEEE.

[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, 11–14 March 2002. Springer-Verlag.

[LL04] Sven Laur and Helger Lipmaa. Additive conditional disclosure of secrets or: Good-bye, random oracles. Submitted, August 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, 1–4 May 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.

[NS98] David Naccache and Jacques Stern. A New Public Key Cryptosystem Based on Higher Residues. In *5th ACM Conference on Computer and Communications Security*, pages 59–66, San Francisco, CA, USA, 3–5 November 1998. ACM Press.

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–318, Helsinki, Finland, May 31 – June 4 1998. Springer-Verlag.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, 2–6 May 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, 18–22 October 1998. Springer-Verlag.