Near Optimal Rate Homomorphic Encryption for Branching Programs

Aggelos Kiayias[1,2], Nikos Leonardos[1], Helger Lipmaa[3], Kateryna Pavlyk[4], and Qiang Tang[1,2]

[1] Department of Informatics and Telecommunications, University of Athens
[2] Department of Computer Science and Engineering, University of Connecticut
[3] Institute of Computer Science, University of Tartu
[4] Institute of Mathematics, University of Tartu

Abstract. We initiate the study of good rate homomorphic encryption schemes. Based on previous work on securely evaluating (binary I/O) branching programs, we propose a leveled homomorphic encryption scheme for *large-output* polynomial-size branching programs (which we call **LBP**) that possesses near optimal-rate. The rate analysis of the new scheme is intricate: the best rate is achieved if a certain parameter s is set equal to the only positive root of a degree-m polynomial, where m is the length of the branching program. We employ the Newton-Puiseux algorithm to find a Puiseux series for this parameter, and based on this, propose a $\Theta(\log m)$-time algorithm to find an integer approximation to s.

We also describe a rate-optimal 1-out-of-n CPIR based on rate-optimal homomorphic encryption. In concrete terms, when applied to say, a movie database with $n = 2^{16}$ elements of $\ell = 3.8 \cdot 10^9$-bits, the client can privately download a movie with a communication rate of almost 0.99, hence sacrificing only about 1% of bandwidth for privacy.

We also analyze the optimality of the rate efficiency of our scheme in a novel model that may be of independent interest. Our 1-out-of-n CPIR has rate $1 - 1.72\sqrt{k/\ell} \cdot \log_2 n + O_\ell(\ell^{-1})$, while we show that no black-box construction surpasses $1 - \sqrt{k/\ell}(\log n / \log\log n) + O_\ell(\ell^{-1})$ in terms of rate, where ℓ is the length of the database elements and k the security parameter.

Keywords: branching programs, CPIR, homomorphic encryption, lower bounds, Puiseux series

1 Introduction

The *rate* of regular public-key encryption scheme $|x|/|\mathsf{Enc}(x)|$ as a function of $\ell = |x|$ is typically of no great concern in cryptography. This is because given any sub-optimal public-key encryption scheme Enc it is possible to transform it to a near optimal one via hybrid encryption. The construction amounts encrypting a key K by using Enc, and then encrypting the plaintext x using a rate-optimal symmetric encryption scheme $\mathsf{SE}(\cdot)$ (which are easy to construct); this results in a near optimal ciphertext $\langle \mathsf{Enc}(K), \mathsf{SE}_K(x) \rangle$.

While the above construction preserves security (assuming a suitable $\mathsf{SE}(\cdot)$), it does not preserve any homomorphic property that $\mathsf{Enc}(\cdot)$ may have. This poses the fundamental question we are concerned with in this work: is it possible to construct a (near) optimal-rate homomorphic public-key encryption? In fact, fully homomorphic encryption schemes (introduced in [Gen09b]) have very low and typically subconstant rate, see [CLO+13] for a recent analysis of the parameters[5]. For rates that are arbitrarily close to 1, the current only documented scheme is due to Damgård and Jurik [DJ01] that allows only an additive homomorphic property, i.e., an evaluation of arithmetic circuits with only an addition gate.[6]

[5] Furthermore, performing hybrid encryption using fully homomorphic encryption $\langle \mathsf{FHE}(K), \mathsf{SE}_K(x) \rangle$ and then homomorphically evaluating AES [GHS12]) does not give any advantage over straight FHE since given f it will merely enable the computation of another FHE ciphertext containing $\mathsf{FHE}(\mathsf{SE}_K(f(x)))$ while our objective in this case is to obtain a ciphertext of the form $\langle \mathsf{FHE}(K), \mathsf{SE}_K(f(x)) \rangle$.

[6] There were occasional claims of a ring-LWE based optimal rate scheme for PIR (e.g., e-print draft [Lip11] that is now withdrawn). However, to the best of our knowledge, no optimal rate PIR or FHE scheme is documented in the published literature; as we exemplify, the performance analysis for optimal rate is involved and we certainly hope that our work will motivate further research in this direction.

Our Contributions. We initiate the study of good rate homomorphic encryption schemes by proposing the first optimal-rate (leveled) homomorphic encryption scheme for a non-trivial family of languages. The new construction is a variation of older constructions; however, the performance analysis needed to optimize various parameters is complicated and seems to be novel in cryptographic research. We show how our construction can be used to optimize the communication of cryptographic tasks such as private information retrieval (PIR) assuming large enough data are to be transferred. We mention several open questions such as achieving optimal rate for larger classes of functions (e.g., via optimizing fully homomorphic encryption) and optimizing the computation of these protocols. We hope that our work gives an impetus to the design of efficient rate public-key cryptographic primitives.

Let C be a class of programs. A C-homomorphic PKE scheme is an encryption scheme where given an encryption $\psi_x = \mathsf{Enc}(x)$ and a program $P_f \in \mathsf{C}$ that implements a function f, it is possible to produce a value $\psi_{f(x)} = \mathsf{Eval}(P_f, \mathsf{Enc}(x))$ that can be decrypted to $f(x)$. It is also required that the output of Eval is succinct and program-private (i.e., it will reveal no information about P_f, [Gen09b]). Such scheme is *leveled* (cf. [Gen09b]) if $\mathsf{Enc}(x)$ should restrict to a subclass of C for which Eval can be applied (e.g., the restriction is typically of the form "programs up to certain length"). Designing homomorphic PKE's for wide classes of programs C is one of the major endeavors in cryptography research, and it has resulted in XOR-circuits stemming from [GM82], addition over $\mathbb{Z}_N$ in [Pai99], **NC1**-homomorphic encryption [SYY99], (**L**/**poly**)-homomorphic encryption [IP07] and finally **P**-homomorphic encryption[Gen09b].

The rate of a C-homomorphic PKE scheme is the ratio $(|x| + |f(x)|)/(|\psi_x| + |\psi_{f(x)}|)$. Here, we include the length of the input x, since x also has to be transported during the protocols, and in many applications $|x|$ is at least as large as $|f(x)|$. The focus on the *rate* of homomorphic PKE's is a novel characteristic of our work.

Let **LBP** be the class of all branching programs with large output[7], that are polynomial-size in terms of their input variables and number of sinks (see Sect. 2 for a precise definition). We construct a leveled **LBP**-homomorphic encryption scheme with optimal rate — i.e., a rate-$(1-1/r(\ell))$, for a rapidly increasing function r. While the existence of such scheme is considered folklore (e.g., it is alluded in [Lip09] but also mentioned informally earlier) no previous work presented a construction or an analysis of its rate. It turns out the design (the choice of optimal parameters) and analysis are both very intricate. In fact, we consider the analysis (that uses powerful techniques from mathematical analysis) of this construction to be one of the main contributions of this work.

The construction is a culmination of a long line of previous schemes that employ additive homomorphic public-key encryption (i.e., G_+-homomorphic encryption where G_+ is the class of all arithmetic circuits over addition gates) and they employ recursion in order to enhance the class of programs that are computable; the technique was introduced by Kushilevitz and Ostrovsky [KO97] and followed (implicitly or explicitly) by a number of works including [Ste98] and [Lip05,IP07,Lip09]. The latter three papers use the high-rate PKE from [DJ01]; nevertheless, none of the previous works provided a way to utilize it in such a way that optimal rate is achieved in the evaluated program. More specifically, we present the following:

Construction. We construct an **LBP**-homomorphic PKE scheme that evaluates efficiently any (leveled) branching program P_f for a function $f : \{0,1\}^\chi \to \{0,1\}^\ell$ where we assume that P_f has length (= the maximum number of levels) m. Our construction has rate $1-2\sqrt{(w-1)\chi mk}\cdot \ell^{-1/2} + O_\ell(\ell^{-1})$, where w is the arity of the branching program and k is the security parameter.

[7] The usual definition of branching programs assumes that the output is binary; in a setting where we are mainly interested in the rate (as a function of $|x| + |f(x)|$), it makes sense to consider very long outputs

It follows the general paradigm of [KO97] as applied in [IP07] to branching program evaluation, with an array of crucial optimizations that are tailored to our goal of achieving optimal rate.

In [Lip05,IP07,Lip09], one recursively applies the cryptosystem of [DJ01]. Every recursion level i defines a length parameter s_i; in the aforementioned constructions the values s_i are strictly increasing. We show that the optimal rate is only achieved if the parameters $s_1, \ldots, s_m$ are suitably selected. We use techniques from multivariate analysis to show that these parameters must be all equal to some value s. After that, we show that the optimal communication results from choosing s as the unique positive root of the degree-$(m+1)$ polynomial $f_m(X, \sigma) := \sigma X^{m+1} - (X+1)^{m-1}$, where $\sigma = (w-1)k\chi/(\ell m)$ for some w (in the case of usual branching programs, $w = 2$). Finding the root is impossible analytically when $m > 3$. Instead, we use the Newton-Puiseux algorithm [Cas00] to compute the Puiseux series $\sum_{i=0}^{\infty} c_i \sigma^{(i-1)/2}$ of the optimal s. We then construct a simple algorithm that, given the first two partial sums of the Puiseux series, computes an integer approximation to the optimal s in $\log_2 m$ steps.

The homomorphic encryption scheme of [IP07] is often dismissed because of its bad communication-efficiency, caused by the fact that the output length of its evaluation algorithm depends on the length of the branching program. Therefore, it is rather surprising that a simple modification like ours allows to achieve optimal rate. The latter becomes clear only after extensive analysis of the parameters as explained above.

Another important aspect of LHE schemes is the server's computation. While this is not a focus of the current work, we remark that the new LHE scheme fares better than [IP07,Lip09] also in this aspect. The reason behind this is that instead of encrypting at least ℓ-bit strings, we encrypt in suitably small segments. Since encryption takes superquadratic time, we therefore save significantly in computation.

Applications. In a CPIR protocol [KO97] a client receives one ℓ-bit block out of n possibilities that form a database maintained by a server. As the first application, we use our results to optimize the communication rate of $(n, 1)$-CPIR. Based on our new leveled **LBP**-homomorphic encryption scheme, we show how to construct an $(n, 1)$-CPIR protocol with communication $\ell + 1.72\sqrt{k\ell}\log_2 n + O(1)$ and optimal rate $1 - 1.72\log_2 n \cdot \sqrt{k/\ell} + O(\ell^{-1})$.[8] Using our lower bound argument (see below) we also establish that any efficient PIR (i.e., one with polylogarithmic communication complexity in n) that is in the black-box additive PKE model and satisfies that the client sends $\Omega(\log n / \log\log n)$ ciphertexts overall (true for all protocols that fit the model [KO97,Lip05,Lip09]) has communication at least $\ell + c \cdot \Omega(\log n / \log\log n) \cdot \sqrt{k\ell}$. We note that CPIR protocols using the ϕ-hiding assumption [CMS99,GR05] are highly unlikely to achieve rate close to 1, see [CMS99, Sect. 2.3] for a discussion. Thus, with standard number theoretic assumptions (our scheme is based on RSA) the results we present are the best possible for the total communication of CPIR up to log log-factors.

We also present concrete parameter choices for the new CPIR protocol, demonstrating that in a practical relevant case (say in an application where a client wants to privately retrieve a 3.8 Gigabyte movie from the server's database of 2^{16} movies) one can get rate 0.99. We emphasize that in practice, the resulting rate 0.99 is obviously of greater value than a theoretical estimate of type $1 - o(1)$; it shows the great practical value of the new **LBP**-homomorphic PKE scheme.

Charpentier et al [CFFC11] observed that oblivious transfer could be used as a building block for asymmetric fingerprinting [Pfi96] directly, in which a client downloads a file (usually with large size, e.g., a movie, thus communication rate is a key factor of efficiency) from a server so

[8] One may argue that in the case of $(n, 1)$-CPIR, it is more important to optimize the server's computational complexity (that is linear in all existing protocols but [Lip09]). Interestingly enough, for large ℓ, the new protocol achieves better *concrete* computational complexity than the protocols from [Lip05,Lip09]. This is since the latter protocols require exponentiations with $\approx \ell$-bit strings (each taking at least $\Theta(\ell^{2+o(1)})$ bit-operations), while in the new protocol, one has to only perform exponentiations with $\approx \sqrt{k\ell}$-bit strings.

that it is *fingerprinted* in a private fashion. In this way, the server obtains undeniable proof of the implication of a client in a piracy incident. All previous works, e.g.,[Pfi96,PS96,PW97] treated the file transfer generically as an instance of secure two party computation. Unfortunately, even with "communication efficient" secure two-party computation [NN01,DFH12,DZ13,IKM+13] the complexity of the resulting protocol is prohibitively high.

As the CPIR application can be easily adapted to OT, our rate optimal homomorphic encryption opens the door to practical asymmetric fingerprinting schemes, and in principle it could be applied to construct an asymmetric fingerprinting scheme with rate close to 1.

Lower Bounds. We prove a lower bound for the communication length of all leveled **LBP**-homomorphic PKE schemes that perform the program evaluation via the utilization of an underlying additively homomorphic PKE scheme in a black-box way. Our communication lower bound for the case of **LBP**-homomorphic PKE gives us a ceiling for the best possible rate that can be achieved in the context of private-information-retrieval which is $1 - \Omega(\frac{\log n}{\log \log n}) \cdot \sqrt{k/\ell} + \Theta(\ell^{-1})$. This demonstrates the optimality of our construction within logarithmic factors. We achieve this result by introducing a model for arguing about lower bounds in the context of secure two-party computation that we call the black-box additive PKE model and may be of independent interest. All previous related constructions [KO97,Lip05,IP07,Lip09] can be described within our model. We show two lower bounds that are complementary: the first one establishes the importance of the term $\sqrt{k/\ell}$ in the ceiling of the rate while the second demonstrates the lower bound dependency of the bound in the number of levels m. The first bound is shown directly with an information theoretic argument that takes into account the need to encode the correct output stream. The second bound is more involved and relies on an intermediate lower bound that examines the communication complexity in a hybrid model where the two parties communicate via a multivariate polynomial evaluation oracle. Using this intermediate result we determine a lower bound that relates the selection parameter domain size n and the total number of levels.

Paper Organization. In section 2 we present preliminaries including notations and the definition of branching programs. In section 3 we present our main construction for the **LBP**-homomorphic encryption scheme. It is described in a black-box fashion over an additively homomorphic PKE with suitable efficiency and length flexibility properties. The scheme is described leaving a number of parameters free which we in turn optimize in section 4; the final theorem about the efficiency of our construction is given in this section together with some concrete parameters from our implementation. In section 5, as an application of the new encryption scheme, we present a near optimal rate CPIR protocol. Finally, in section 6 we describe our lower bound results.

Due to the lack of space, many technical details are given in the appendix.

2 Preliminaries

Notation. We use aggressively the Landau notation like $\Theta(\cdot)$. Since we are often interested in the growth of a function in several variables, we write the relevant variable as a subscript, like in $o_\ell(\ell \log n)$. Let k be the polynomial security parameter, i.e., we assume that adversaries work in probabilistic polynomial-time w.r.t. k. The current recommendation is to take $k \geq 3072$. If not specified, all logarithms take basis 2; we denote the natural logarithm of x by $\ln x$. For a predicate $P(x)$, let $[P(x)] = 1$ if $P(x)$ is true, and $= 0$, otherwise. We occasionally use the notation $[n]$ for the integer set $\{1, \ldots, n\}$.

(Large Output) Branching Programs. A *w-ary branching program* (if $w = 2$, it is more commonly known as a binary decision diagram or BDD, [Weg00]) is a fanout-w directed acyclic

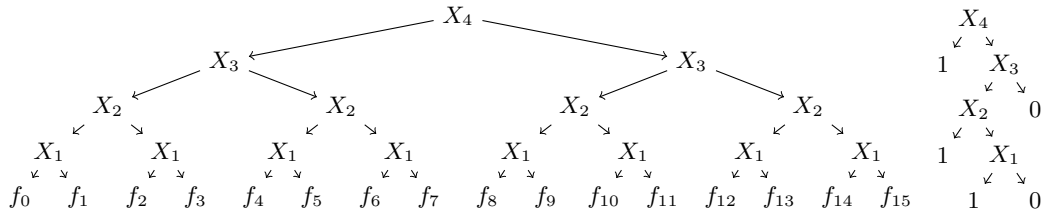

Fig. 1. The complete binary decision tree that returns f_x (left), and a branching program that returns 1 iff $x \leq 10$ (right). In both cases, $\mathsf{len}(P) = 4$

graph (V, E), where the non-terminal (that is, non-sink) nodes are labeled by variables from some variable set $\{X_1, \ldots, X_\chi\}$, the sinks are labeled by ℓ-bit strings and the w outgoing edges of every internal node are labeled by values from 0 to $w-1$. Usually, it is assumed that a branching program has 1-bit sink labels, then it can be assumed to have two terminal nodes. A branching program with longer sink labels is thus sometimes called *multi-terminal*. See Fig. 1.

A branching program with 1 source computes a function $f : \{0,1\}^\chi \to \{0,1\}^\ell$. Every assignment of the variables selects one path from the source to a sink as follows. The path starts from the source. If the current path does not end at a sink, test the variable at the endpoint of the path. Select one of the outgoing edges depending on the value of this variable, and append this edge and its endpoint to the path. (For the sake of concreteness, we assume that the leftmost edge is chosen iff the variable is 0.) If the path ends at a sink, return the label of this sink as the value of the source. The branching program's value is then equal to the source value.

For a branching program P, let $\mathsf{len}(P)$ be its length (that is, the length of its longest path), $\mathsf{size}(P)$ be its size (that is, the number of non-terminal nodes). Let $\mathsf{BP}(f)$ be the minimal size of any branching program computing f. It is known that any Boolean function $f : \{0,1\}^\chi \to \{0,1\}$ has $\mathsf{BP}(f) \leq (1 + o(1))2^\chi/\chi$ [BHR95, Thm. 1]. A Boolean function f has a polynomial-size branching program iff f belongs to $\mathbf{L/poly}$ [Cob66]. If f has non-Boolean output, $f : \{0, \ldots, w-1\}^\chi \to \{0,1\}^\ell$, then it can be computed in parallel by ℓ branching programs that compute its individual bits. P is a *decision tree* if the underlying graph is a tree. P is *leveled* if its set of nodes can be divided into disjoint sets V_d such that every edge from a node in set V_d ends in a node in set V_{d-1}. We assume that the source is the only member of the set V_m. Let $\mathsf{size}(P, d)$ be the number of nodes P has on level d, thus $\mathsf{size}(P, m) = 1$.

Definition 1. *The class* **LBP** *contains all functions* $f : \{0, \ldots, w-1\}^\chi \to \{0,1\}^\ell$ *for which we have a large-output branching program with size that is polynomial on both parameters* χ, n *with* $n = |f(\{0, \ldots, w-1\}^\chi)|$.

Throughout the paper, P_f is a fixed leveled w-ary branching program that implements $f : \{0,1\}^\chi \to \{0,1\}^\ell$. For any node v, let $\mathsf{len}(v)$ be its *length*, i.e., $\mathsf{len}(v) = \mathsf{len}(P_f) - \bar{d}$, where $\bar{d}$ is the distance from the source to v. Thus $v \in V_{\mathsf{len}(v)}$, and the source has length $\mathsf{len}(P_f)$. (E.g., on Fig. 1 (left), all sinks have length 0 and the source has length 4.) If v is a non-sink node, let $\mathsf{child}(v, i)$ for $i \in [0, w-1]$ be its ith leftmost child, and $X_{\mathsf{ind}(v)}$ be the label of v. Note that $\mathsf{len}(\mathsf{child}(v, i)) = \mathsf{len}(v) - 1$. Assume that the nodes of P_f are ordered from 1 to $\mathsf{size}(P_f)$ so that if there exists an edge $u \to v$ then $v < u$. We assume that P_f has n sinks. Thus, nodes $v \leq n$ are the sinks and $v = \mathsf{size}(P_f)$ is the source of P_f. Recall that the description of P_f also contains the values $\boldsymbol{f_v}$ of the sinks of P_f.

Public-Key Cryptosystem. A public-key cryptosystem Π consists of three algorithms, a probabilistic polynomial-time key generating algorithm $(\mathsf{pk}, \mathsf{sk}) \leftarrow_r \mathsf{Gen}(1^k)$, a probabilistic

polynomial-time encryption algorithm $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(x;r)$, and a deterministic polynomial-time decryption algorithm $x \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c)$. It is required that if $(\mathsf{pk},\mathsf{sk}) \leftarrow_r \mathsf{Gen}(1^k)$, then for any x and r from corresponding domains, $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(x;r)) = x$. The *rate* of a cryptosystem is the length ℓ of a plaintext divided by the length of its ciphertext. The rate can be a function of ℓ. A cryptosystem is *CPA-secure* if for any x_0 and x_1 (possibly chosen by the adversary) of the same length, given an encryption $\mathsf{Enc}^s_N(x_\beta;r)$ for randomly chosen $\beta \in \{0,1\}$ and r, no probabilistic polynomial-time adversary can guess β with probability $1/2+\varepsilon$, where ε is non-negligible in k.

Damgård-Jurik Cryptosystem. Assume ℓ is the length of the plaintexts in bits. The Damgård-Jurik cryptosystem [DJ01] allows to encrypt plaintexts for arbitrary $\ell \geq 1$, so that the ciphertext length is not more than $\ell + 2k$. The cryptosystem is defined as follows. To generate the public and secret keys, one lets $N = pq$ to be a k-bit RSA modulus for two randomly generated large $k/2$-bit primes p and q. The value N is the public key pk, and the factorization (p,q) of N is a part (together with some additional information that makes decrypting more efficient) of the secret key sk. To encrypt an ℓ-bit string x, one chooses *a length parameter* s such that $\ell = sk$ (or $s = \lceil \ell/k \rceil$ if $k \nmid \ell$), chooses a randomizer $r \leftarrow_r \mathbb{Z}^*_N$, and then outputs $c \leftarrow (1+N)^x r^{N^s} \mod N^{s+1}$. Thus the plaintext belongs to $\mathbb{Z}_{N^s}$ while the ciphertext belongs to $\mathbb{Z}_{N^{s+1}}$, i.e., has the bitlength $\leq \lceil \log_2 N^{s+1} \rceil \leq (s+1)k$ bits. Due to the choice of s, the bitlength of the plaintext is at least $(s-1)k$. Decryption can be done efficiently, see [DJ01].

The *rate* of Damgård-Jurik is $|x|/|c| \geq (s-1)/(s+1)$. If $\ell \to \infty$, then $\ell = |x| \approx sk$, and the rate is approximately $1 - 1/s$. Since $\mathsf{Enc}^s_N(x_0;r_0) \cdot \mathsf{Enc}^s_N(x_1;r_1) = \mathsf{Enc}^s_N(x_0+x_1;r_0r_1)$, the Damgård-Jurik cryptosystem is additively homomorphic. If c is a publicly known value, then $\mathsf{Enc}^s_N(x;r)^c = \mathsf{Enc}^s_N(cx;r^c)$. Recall that arithmetic in the first (resp., second) parameter of Enc is done modulo N^s (resp., N).

The CPA-security of the Damgård-Jurik cryptosystem is based on the Decisional Composite Residuosity (DCR) assumption of Paillier [Pai99].

Computationally-Private Information Retrieval (CPIR). In an $(w,1)$-CPIR protocol for ℓ-bit strings, the server has a database of w elements, $\boldsymbol{f} = (f_0,\ldots,f_{w-1})$, each f_i being ℓ bits long, and the client has an input $x \in \{0,\ldots,w-1\}$. The client needs to obtain f_x, while no efficient (i.e., probabilistic polynomial-time) server should obtain any information about x. In a *two-message* CPIR protocol, the client generates a secret/public key pair $(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{KGCPIR}(1^k)$, and sends to the server pk and a query $c = \mathsf{Que}_{\mathsf{pk}}(w,\ell,x;r)$ that depends on the security parameter k, the size of the database w, the length of the database elements ℓ, the input x, and some random coins r. The server replies with $\hat{c} \leftarrow \mathsf{Rep}_{\mathsf{pk}}(\boldsymbol{f},c;\hat{r})$ that depends on the input $\boldsymbol{f}$, the query c, and another randomizer $\hat{r}$. The client can recover f_x by using algorithm $\mathsf{Ans}_{\mathsf{sk}}(w,\ell,\hat{c})$, given access to $\hat{c}$, w, ℓ, and the secret key sk.

The *rate* $\mathsf{rate}(\Gamma)$ of a (two-message) CPIR protocol Γ is the number of "useful bits" (that is, ℓ, the database element length) divided by the total communication $|\mathsf{Que}()| + |\mathsf{Rep}()|$ of the protocol. We do not include pk to the communication, since the same pk can — and will — be used in many CPIR protocols.

The formal (CPA-)security notion is similar to the one of cryptosystems, see, e.g., [Lip05].

Lipmaa's Basic $(w,1)$-CPIR. In Lipmaa's $(w,1)$-CPIR protocol [Lip05,Lip09], the client first generates secret and public key $(\mathsf{sk},\mathsf{pk})$, with $\mathsf{pk} = N$, for the Damgård-Jurik cryptosystem. She then sends pk together with a vector of $w-1$ ciphertexts

$$\boldsymbol{c} = (c_1,\ldots,c_{w-1}) = \mathsf{Que}_{\mathsf{pk}}(w,\ell,x;\boldsymbol{r}) \leftarrow (\mathsf{Enc}^s_N([i=x];r_i))_{i=1}^{w-1}$$

to the server, where $[i=x] \in \{0,1\}$ is equal to 1 iff $i = x$, $s = \lceil \ell/k \rceil$, $r_i \leftarrow_r \mathbb{Z}^*_N$, and $\boldsymbol{r} = (r_1,\ldots,r_{w-1})$. Note that $|c_i| = (s+1)k$ and thus $|\boldsymbol{c}| = (w-1)(s+1)k$. Let $\boldsymbol{f} = (f_0,\ldots,f_{w-1})$

be the server's database. The server answers, for random $\hat{r} \leftarrow_r \mathbb{Z}_N^*$, with

$$\hat{c} \leftarrow \mathsf{Rep}_{\mathsf{pk}}(\boldsymbol{f}, \boldsymbol{c}; \hat{r}) := \prod_{i=1}^{w-1} c_i^{f_i - f_0} \cdot \mathsf{Enc}_N^s(f_0; \hat{r}) = \mathsf{Enc}_N^s(f_x; \prod_{i=1}^{w-1} r_i^{f_i - f_0} \cdot \hat{r}) \ .$$

Since $\hat{r}$ is random, then $\hat{c}$ is a random encryption of f_x.

The client obtains f_x by decrypting $\hat{c}$, $f_x \leftarrow \mathsf{Ans}_{\mathsf{sk}}(w, \ell, \hat{c}) := \mathsf{Dec}_{\mathsf{sk}}^s(\hat{c})$. Clearly, the server's answer is a random encryption of f_x. Since the server only sees encrypted messages, the CPA-security of $(w, 1)$-Lipmaa's CPIR protocol immediately follows from the CPA-security of the Damgård-Jurik cryptosystem, and thus, from the DCR assumption.

Here, $|\mathsf{Que}()| = (w-1)(s+1)k$ and $|\mathsf{Rep}()| = (s+1)k$, where $s = \lceil \ell/k \rceil$, and thus the rate is $\mathsf{rate}(\Gamma) = (sk)/((w-1)(s+1)k) = s/((w-1)(s+1)) = 1/(w-1) - k/((w-1)\ell) + O_\ell(\ell^{-2})$, since $s \approx \ell/k$.

Due to the construction of the Damgård-Jurik cryptosystem, x and f_x must be encrypted by using the same length parameter s: if x was encrypted by using a parameter $z < s$, then the server's answer would encrypt $f_x \mod N^z$ and thus the server would not recover the whole value f_x. More discussion on this issue is provided in [Lip09]. There it was also demonstrated that (i) There exists a function $\mathsf{Compress}$ that takes $\mathsf{Enc}_N^{s+1}(x; r)$ as an input and outputs $\mathsf{Enc}_N^s(x \mod N^s; r^*)$, where $s \geq 1$, and r^* is a randomizer that depends on N, r and s. (ii) On the other hand, if there exists a function Expand that takes $\mathsf{Enc}_N^s(x; r^*)$ as an input and outputs $\mathsf{Enc}_N^{s+1}(x; r)$, then the Lipmaa's $(2, 1)$-CPIR protocol has rate $1 - o(1)$. Namely, in this case the client's message is just $\mathsf{Enc}_N^1(x)$ ($2k$ bits) independently of the value of ℓ. However, the existence of this function was deemed to be extremely unlikely in [Lip09].

3 Optimal-Rate Leveled LBP-Homomorphic Encryption

We introduce *leveled* homomorphic encryption for **LBP**, following the terminology of Gentry [Gen09a]. However, the definition will be somewhat different. According to [Gen09a, Def. 2.1.5], a family of homomorphic encryption schemes $\{\Pi^{(m)} : m \in \mathbb{Z}^+\}$ is *leveled fully homomorphic* if, for all $m \in \mathbb{Z}^+$, they all use the same decryption circuit, $\Pi^{(m)}$ compactly evaluates all circuits of depth at most m (that use some specified set of gates), and the computational complexity of $\Pi^{(m)}$'s algorithms is polynomial in k, m, and (in the case of the evaluation algorithm) the size of the circuit. In practice, this means that each $\Pi^{(m)}$ can have a different private/public key pair $(\mathsf{sk}^{(m)}, \mathsf{pk}^{(m)})$. This is since the public moduli, used when encrypting, (and thus also the public key) depend on m.

The new (slightly stronger) definition only requires the creation of a single key pair $(\mathsf{sk}, \mathsf{pk})$ usable for any m. Instead, it is Alice who picks the value m while encrypting the messages. The concrete value m gives an *upper bound* on the length of the large-output branching program that Bob can evaluate on these ciphertexts. Optimal rate is achieved if m is equal to the actual length of the evaluated large-output branching program. For this reason, in the definition we will concentrate on the case of level m branching programs. Since the rate in our case will be defined as the total length of Alice's and Bob's messages, it is is natural that Alice — who sends her message first — has to choose the parameter m, based on her knowledge of Bob's input, to optimize the rate. Similar problem exists in leveled FHE.

Definition 2. *A* (single-key) leveled **LBP**-homomorphic encryption (LHE) scheme *is a four-tuple of efficient algorithms* $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$*, such that (i) the randomized* key generation algorithm $\mathsf{KG}(1^k)$ *creates a single secret key and public key pair* $(\mathsf{sk}, \mathsf{pk})$*, (ii) given a message* x*, the branching program length* m*, and a randomizer* r*, the* randomized encryption algorithm

$\mathsf{Enc}^m_{\mathsf{pk}}(x;r)$ *returns a ciphertext* c*, (iii) given a leveled branching program* P_f *of length* m*, a fresh ciphertext* c *(equal to* $\mathsf{Enc}^m_{\mathsf{pk}}(x;r)$ *for some plaintext* x *and randomizer* r*) and a randomizer* $\hat{r}$*, the* randomized evaluation algorithm $\mathsf{Eval}_{\mathsf{pk}}(P_f, c; \hat{r})$ *returns an evaluated ciphertext* $\hat{c}$ *of* $P_f(x)$*, (iv) given an evaluated ciphertext* $\hat{c}$ *and the branching program length* m*, the* deterministic decryption algorithm $\mathsf{Dec}^m_{\mathsf{sk}}(\hat{c})$ *returns a plaintext* x*.*

It is required that for any valid key pair $(\mathsf{sk}, \mathsf{pk})$, message x, randomizers r and $\hat{r}$, and a polynomial-size branching program P_f of length m, $\mathsf{Dec}^m_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{pk}}(P_f, \mathsf{Enc}^m_{\mathsf{pk}}(x;r); \hat{r})) = x$. Finally, it must be the case that the computational complexity of these four algorithms is polynomial in k, m, and (in the case of Eval) the size of the branching program. We often omit the randomizer r (or $\hat{r}$) in the descrption of algorithms; in this case it is understood that it is chosen uniformly at random.

A leveled LHE scheme must satisfy two security requirements, *CPA-security* and *branching program privacy* (similar to circuit-privacy, [Gen09b,Gen09a]). The first one is defined similarly to the case of arbitrary public-key cryptosystems, though one has to take into account the presence of Eval, see [Gen09b,Gen09a] for a formal definition. However, to achieve optimal rate we allow the outputs of Enc^m and Eval can come from different distributions; we just require that the output of Eval does not reveal any unnecessary information about the evaluated branching program except its length. That is, *(perfect) branching program privacy* guarantees that for any $(\mathsf{sk}, \mathsf{pk})$, any m, any valid ciphertext c produced by $\mathsf{Enc}^m_{\mathsf{pk}}$, and any two equal-length branching programs P_0 and P_1 such that $P_0(\mathsf{Dec}^m_{\mathsf{sk}}(c)) = P_1(\mathsf{Dec}^m_{\mathsf{sk}}(c))$, it holds that $\mathsf{Eval}_{\mathsf{pk}}(P_0, c)$ and $\mathsf{Eval}_{\mathsf{pk}}(P_1, c)$ have the same distribution.

We require that the LHE scheme Π be communication-efficient in the sense that its *rate*

$$\mathsf{rate}(\Pi) := \frac{|x| + |P_f(x)|}{|\mathsf{Enc}^m_{\mathsf{pk}}(x;r)| + |\mathsf{Eval}_{\mathsf{pk}}(P_f, \mathsf{Enc}^m_{\mathsf{pk}}(x;r); \hat{r})|}$$

is as large as possible. Informally, Π is *optimal-rate*, if the rate is $1 - o_\ell(1)$ as a function of ℓ. The rate takes into account the value $|\mathsf{Enc}|$, since it is possible to choose parameters so that $|\mathsf{Eval}|$ is very small while $|\mathsf{Enc}|$ is very large. It is also a natural measurement of the rate in many applications like $(n, 1)$-CPIR. Similarly, the *communication complexity* of a leveled LHE scheme is equal to $|\mathsf{Enc}| + |\mathsf{Eval}|$.

Construction. Next, we propose a leveled LHE scheme that enables one to securely compute the value of any function $f : \{0, \ldots, w-1\}^\chi \to \{0,1\}^\ell$, $f \in \mathbf{LBP}$, with the communication complexity that is approximately $\ell + 2\sqrt{(w-1)\chi mk\ell}$ (+ smaller terms), and the rate that is $1 - 2\sqrt{(w-1)\chi mk/\ell}$ (+ smaller terms). Here, m is the length of a polynomial-size w-ary leveled branching program P_f that implements f. Since in the intended applications, $\ell \gg \chi mk$, the rate will quickly approach 1 when ℓ increases.

We utilize a two-message $(w, 1)$-CPIR protocol (see Sect. 2) with short reply $|\mathsf{Rep}()|$. More precisely, we chose Lipmaa's $(w, 1)$-CPIR $\Gamma = (\mathsf{KGCPIR}, \mathsf{Que}, \mathsf{Rep}, \mathsf{Ans})$ (see Sect. 2).[9] We recall that in the case of this CPIR, $|\mathsf{Que}()| = (w-1)(s+1)k$ and $|\mathsf{Rep}()| = (s+1)k$, where $s = \lceil \ell/k \rceil$. Furthermore, it is possible to derive $\mathsf{Que}_{\mathsf{pk}}(w, \ell', x; \cdot)$ from $\mathsf{Que}_{\mathsf{pk}}(w, \ell, x; r)$ for any $\ell' \le \ell$ without knowing the secret key [Lip09] (see Sect. 2).

We assume that all parties know $m = \mathsf{len}(P_f)$. For every level $d \in [m]$, let s_d be a level-specific length parameter. Optimal parameters s_d, for $d \in [m]$, will be fixed in Sect. 4. Every node v has a label $\mathfrak{L}_v$ of bitlength $|\mathfrak{L}_v| = t_{\mathsf{len}(v)} s_{\mathsf{len}(v)} k$ for some values t_i defined later. Let $s_i^{\max} := \max\{s_d :$

[9] In fact, we considered a much wider class of $(w, 1)$-CPIR protocols, proposed in [Lip09], that have a better rate than Γ. However, it came out that the optimal case corresponds to Γ. Briefly, the reason is that in the current paper we need Γ to have an extremely short $\mathsf{Rep}()$, while the length of $\mathsf{Que}()$ is not so essential. To not make this paper even more longer, we have omitted further discussion.

some level d node is labeled by X_i} Sometimes, but not always, it is reasonable to assume that the encrypter knows the values $s_i^{\max}$. If he does not, we can assume that $s_i^{\max} := \max\{s_d\}$. In the optimal case, see Sect. 4, all values s_d are equal to each other, so it does not matter whether the encrypter knows the values $s_i^{\max}$. However, we first have to establish the optimality.

On input x, the encrypter writes $x = \sum x_i w^i$ with $x_i < w$, and then for each x_i provides $\boldsymbol{c_i} \leftarrow \mathsf{Que}_{pk}(w, s_i^{\max} \cdot k, x_i; \cdot)$ by using a new randomness. The vector of those queries is the LHE encryption of x. Note that x_i corresponds to an assignment to the formal variable X_i.

$\mathsf{Eval}_{\mathsf{pk}}(P_f, c; \hat{r})$ inputs a w-ary leveled branching program P_f and the queries $\boldsymbol{c_i}$ corresponding to assignments to all X_i. Recall that the choice of P_f fixes $\mathfrak{L}_{\boldsymbol{v}}$ for all sinks $v \le n$. Eval then recursively computes $\mathfrak{L}_{\boldsymbol{v}}$ for all non-sink nodes whose children already have assigned labels $\mathfrak{L}_{\mathsf{child}(\boldsymbol{v},\boldsymbol{i})}$. Eval returns the value $\mathfrak{L}_{\mathsf{size}(\boldsymbol{P_f})}$ of the source.

Up to now, the construction does not different from those in [IP07,Lip09]. The crux of the new construction is in how exactly $\mathfrak{L}_{\boldsymbol{v}}$ is evaluated. Namely, at every non-sink node v, Eval does the following. (See Fig. 2.) Let $d = \mathsf{len}(v)$; since v is a non-sink node, $\{1, \ldots, m\}$. Recall that $\mathfrak{L}_{\boldsymbol{v}}$ has bit-length $t_d s_d k$ and $\mathfrak{L}_{\mathsf{child}(\boldsymbol{v},\boldsymbol{i})}$ has bit-length $t_{d-1}(s_{d-1}+1)k$. For the evaluation to succeed, we set recursively

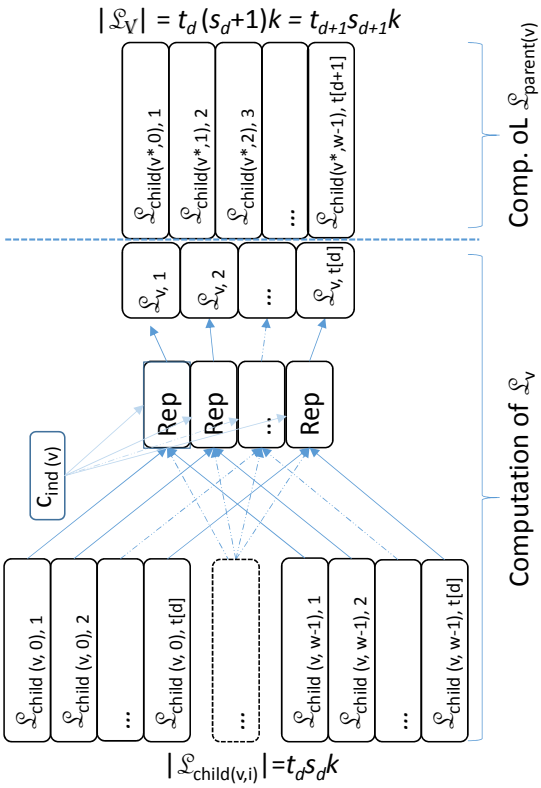

Fig. 2. Local computation of $\mathfrak{L}_{\boldsymbol{v}}$

$$t_0 s_0 k = \ell \ , \quad \text{and} \quad t_d s_d k = t_{d-1}(s_{d-1}+1)k \quad \text{for } d \in [1, m] \ . \tag{1}$$

Eval writes $\mathfrak{L}_{\mathsf{child}(\boldsymbol{v},\boldsymbol{i})} = (\mathfrak{L}_{\mathsf{child}(v,i),1}, \ldots, \mathfrak{L}_{\mathsf{child}(v,i),t_d})$ and $\mathfrak{L}_{\boldsymbol{v}} = (\mathfrak{L}_{v,1}, \ldots, \mathfrak{L}_{v,t_d})$, where $|\mathfrak{L}_{\mathsf{child}(v,i),z}| = s_d$ and due to the properties of the underlying $(w,1)$-CPIR protocol $|\mathfrak{L}_{v,z}| = |\mathsf{Rep}_{\mathsf{pk}}()| = s_d + 1$. (Later, when v will play the role of a child of some other node v^*, $\mathfrak{L}_{\boldsymbol{v}}$ will be divided into t_{d+1} parts, see Fig. 2.) For each z, Eval then computes $\mathfrak{L}_{v,z}$ by applying the $\mathsf{Rep}_{\mathsf{pk}}(\cdot)$ algorithm on $(\mathfrak{L}_{\mathsf{child}(v,i),z})_{i=0}^{w-1}$ and $\boldsymbol{c}_{\mathsf{ind}(\boldsymbol{v})}$, see Fig. 3. After that, $\mathfrak{L}_{\boldsymbol{v}}$ is set equal to the concatenation of the replies of t_d parallel CPIR protocols, where the zth $(w,1)$-CPIR protocol is applied to client's input $x_{\mathsf{ind}(v)}$ and the database $(\mathfrak{L}_{\mathsf{child}(v,i),z})_{i=0}^{w-1}$. Intuitively, $\mathfrak{L}_{\boldsymbol{v}}$ is a "garbled" version of $\mathfrak{L}_{\mathsf{child}(\boldsymbol{v},\boldsymbol{x}_{\mathsf{ind}(\boldsymbol{v})})}$.

This means that at the end of the protocol, $\mathfrak{L}_{\mathsf{size}(\boldsymbol{P_f})}$ is equal to the m-times recursive (and parallel) application of Rep to f_x. From this, the decrypter can obtain f_x from $\mathfrak{L}_{\mathsf{size}(\boldsymbol{P_f})}$ by recursively applying $\mathsf{Ans}_{\mathsf{sk}}$ to it. In our case, $\mathfrak{L}_{\mathsf{size}(\boldsymbol{P_f})}$ (and the intermediate values) is interpreted as a concatenation of t_d bitstrings, and $\mathsf{Ans}_{\mathsf{sk}}$ is applied to each piece separately. The answers are concatenated again, and the result is given as an input to $\mathsf{Ans}_{\mathsf{sk}}$ of the next level. The

Key generation $\mathsf{KG}(1^k)$**:** generate a key pair $(\mathsf{sk}, \mathsf{pk})$ via $\mathsf{KGCPIR}(1^k)$.
Encryption $\mathsf{Enc}^m_{\mathsf{pk}}(x;\cdot)$**:** For $i \leftarrow 1$ to χ, let $\boldsymbol{c_i} \leftarrow \mathsf{Que}_{\mathsf{pk}}(w, s_i^{\max}\cdot k, x_i; \boldsymbol{r}_i)$ for random $\boldsymbol{r}_i$. Return $c \leftarrow (\boldsymbol{c_1}, \ldots, \boldsymbol{c_\chi})$.
Evaluation $\mathsf{Eval}_{\mathsf{pk}}(P_f, c; \hat{r})$ **for** w**-ary branching program** P_f **where** $f : \{0,\ldots,w-1\}^\chi \to \{0,1\}^\ell$**:**

for $v \leftarrow n+1$ **to** $\mathsf{size}(P_f)$ **do**
  $d \leftarrow \mathsf{len}(v)$;
  Parse $\boldsymbol{c_{\mathsf{ind}(v)}}$ as $\boldsymbol{c_{\mathsf{ind}(v)}} = \mathsf{Que}_{\mathsf{pk}}(w, s_i \cdot k, x_i; \cdot)$;
  for $z \leftarrow 1$ **to** t_d **do**
    $\mathfrak{L}_{v,z} \leftarrow \mathsf{Rep}_{\mathsf{pk}}(\mathfrak{L}_{\mathsf{child}(v,0),z}, \ldots, \mathfrak{L}_{\mathsf{child}(v,w-1),z}; \boldsymbol{c_{\mathsf{ind}(v)}}; \hat{r}_{v,z})$ for random $\hat{r}_{v,z}$;
  end
  $\boldsymbol{\mathfrak{L}_v} \leftarrow (\mathfrak{L}_{v,1}, \ldots, \mathfrak{L}_{v,t_d})$;
end
return $\hat{\boldsymbol{c}} \leftarrow \boldsymbol{\mathfrak{L}_{\mathsf{size}(P_f)}}$;

Decryption $\mathsf{Dec}^m_{\mathsf{sk}}(\hat{\boldsymbol{c}})$**:**

Parse $\hat{\boldsymbol{c}} \leftarrow (\hat{c}_1, \ldots, \hat{c}_{t_d})$;
for $z \leftarrow 1$ **to** t_d **do** let $x_z \leftarrow \mathsf{Ans}_{\mathsf{sk}}(w, s_m \cdot k, \hat{c}_z)$;
Write $\boldsymbol{x} = (x_1, \ldots, x_{t_d})$;
if $d = 0$ **then return** $\boldsymbol{x}$;
else return $\mathsf{Dec}^{m-1}_{\mathsf{sk}}(\boldsymbol{x})$;

Fig. 3. The new leveled LHE scheme based on a $(w,1)$-CPIR protocol $(\mathsf{KGCPIR}, \mathsf{Que}, \mathsf{Rep}, \mathsf{Ans})$.

algorithms $\mathsf{KG}(1^k)$, $\mathsf{Enc}^m_{\mathsf{pk}}(x;r)$, $\mathsf{Eval}_{\mathsf{pk}}(P_f, c; \hat{r})$, and $\mathsf{Dec}^m_{\mathsf{sk}}(\hat{c})$ are formally described by Fig. 3. The required constraints are satisfied by Lipmaa's $(w,1)$-CPIR, see Sect. 2.

Theorem 1. *Let* $\Gamma = (\mathsf{KG}, \mathsf{Que}, \mathsf{Rep}, \mathsf{Ans})$ *be a* $(w,1)$*-CPIR protocol that has the* Compress *function, with plaintext size* $\ell = sk$*,* $|\mathsf{Que}(w, \ell, \ldots)| = (w-1)(s+1)k$ *and* $|\mathsf{Rep}()| = (s+1)k$*, where* k *is the security parameter. Let* $f : \{0,\ldots,w-1\}^\chi \to \{0,1\}^\ell$ *be a function in* **LBP***. Let* P_f *be a polynomial-size leveled* w*-ary branching program that implements* f*, and let* $m = \mathsf{len}(P_f)$*. Write* $\boldsymbol{s} = (s_1, \ldots, s_m)$*.*

Let $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *be the leveled LHE scheme from Fig. 3 parameterized by* $w \in \mathbb{N}^{>0}$*.* Π *is perfectly branching program private. If* Γ *is CPA-secure, then* Π *is CPA-secure. The computation of* $\mathsf{Eval}_{\mathsf{pk}}$ *is dominated by* $\sum_{d=1}^m t_d \cdot \mathsf{size}(P_f, d) \cdot T_{\mathsf{Rep}}(s_d, w)$*, where* $T_{\mathsf{Rep}}(s_d, w)$ *is the computational complexity of* $\mathsf{Rep}_{\mathsf{pk}}$ *with given parameters. The communication* $|\mathsf{Enc}| + |\mathsf{Eval}|$ *of* Π *is equal to*

$$\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell) = (w-1)k\left(\sum_{i=1}^{\chi} s_i^{\max} + \chi\right) + \ell \cdot \prod_{d=0}^{m-1}\left(1 + \frac{1}{s_d}\right) \quad . \tag{2}$$

Proof. Correctness and Security. The correctness of the construction is obvious. The branching program privacy is clear, since the decrypter only sees an a number of $(m-1)$-times application of Rep to an output of Que, and it is guaranteed by the definition of privacy that the input to the query does not depend on the branching program.

Next, if an adversary is able to break the CPA-security of the leveled LHE scheme, then via a standard hybrid argument she is also able to break the CPA-security of the underlying CPIR protocol.

Complexity. The computational complexity is obvious. For the communication complexity, clearly, $|\mathsf{Enc}^m_{\mathsf{pk}}(x;r)| = \sum_{i=1}^{\chi}(w-1)(s_i^{\max}+1)k = (w-1)(\sum_{i=1}^{\chi} s_i^{\max} + \chi)k$ bits. For $d > 0$,

$$t_d = \frac{s_{d-1}+1}{s_d} t_{d-1} = \prod_{i=0}^{d-1} \frac{s_i+1}{s_{i+1}} \cdot t_0 = \prod_{i=0}^{d-1}(s_i+1) \cdot \prod_{i=1}^{d} \frac{1}{s_i} \cdot \frac{\ell}{s_0 k} = \frac{\ell}{(s_d+1)k} \cdot \prod_{i=0}^{d}\left(1+\frac{1}{s_i}\right) \quad .$$

Thus, $|\mathsf{Eval}_{\mathsf{pk}}(P_f, c; \hat{r})| = t_m s_m k = \ell/((s_m+1)k) \cdot \prod_{i=0}^{m} (1+1/s_i) \cdot s_m k = \ell \cdot \prod_{i=0}^{m-1} (1+1/s_i)$. This gives the claimed communication complexity. □

To guarantee branching program privacy in the case of a malicious encrypter, the encrypter must accompany his ciphertexts with standard zero knowledge proofs that they individually encrypt a Boolean value, and that their sum encrypts 1. This is out of the scope of the current paper. We remark that one can implement every language from $\mathbf{NC^1}$ by using a width-5 branching program [Bar86]. Since then one only has to keep a small constant number of branching program nodes in memory, by using the new leveled LHE scheme, one can implement languages from $\mathbf{NC^1}$ space-efficiently. We omit further discussion.

4 Finding Rate-Optimal Parameters

Next, we find the optimal parameters that result in the best possible rate for the leveled LHE scheme from Sect. 3. More precisely, our goal is to find optimal length parameters s_d, as a function of ℓ. As we will see, this optimization problem has quite an unexpected solution, we now briefly summarize our strategy. First, we show by using standard methods of multivariate analysis that the communication is minimized when the length parameters s_d used at every level are all equal, $s_1 = \cdots = s_m =: s$. Second, we show that optimal s is defined as the unique positive root of a certain degree-$(m+1)$ polynomial. Third, since there is no general algebraic solution to this polynomial (except for $m < 4$), we find a Puiseux series for the unique positive root s (and also for the communication and rate, given optimal s). Fourth, we describe an efficient $\log m$-time algorithm to find an integer approximation for the optimal value s. As we will show, this results in rate that is very close to 1 in practically relevant scenarios.

Rewording the Optimization Problem. In App. A, we show that $\partial\mathsf{com}/\partial s_1 = \cdots = \partial\mathsf{com}/\partial s_m = 0$. Since all s_i are positive, then the global minimum is reached if $s_1 = \ldots = s_m =: s$ for *some* s. (See App. A for a precise proof.) In particular, this means that the optimal communication complexity does not depend on the fact whether the encrypter knows the values $s_i^{\max}$. Since all s_d-s are equal, we denote

$$\mathsf{com}(\chi, w, m, s, k, \ell) := \mathsf{com}(\chi, w, m, s, \ldots, s, k, \ell) = (w-1)\chi(s+1)k + \left(1+\frac{1}{s}\right)^m \cdot \ell \ . \quad (3)$$

Now, $\partial\mathsf{com}/\partial s = (w-1)\chi k - m(s+1)^{m-1}/s^{m+1} \cdot \ell$. Denoting

$$\sigma := \frac{(w-1)\chi k}{m\ell} \ , \quad (4)$$

the requirement $\partial\mathsf{com}/\partial s = 0$ is equivalent to the requirement that (s, σ) is a root of

$$f_m(x, y) := yx^{m+1} - (x+1)^{m-1} \ . \quad (5)$$

According to the Descartes' rule of signs, f_m has exactly one positive real root for each $m > 0$. Thus, this unique positive real root s also minimizes the function com.

Computing Puiseux Series of s. Since by the Abel-Ruffini theorem there is no general algebraic solution to polynomial equations of degree five or higher, for $m > 3$ it is impossible to give explicit algebraic formula of the root of (5). We tackle this problem as follows.

Recall that the *Puiseux series* [Cas00] of a function $g(x)$ is of type $g(x) = \sum_{i=0}^{\infty} a_i x^{i/\gamma}$, where γ is an integer. We use the Newton-Puiseux algorithm [Cas00] to find the Puiseux series for the unique positive root s of f_m. By the previous discussion, this will also be the Puiseux series for the value of s that minimizes com.

First, it is known [Cas00] that the Puiseux series exists, i.e., $s = \sum_{i=0}^{\infty} a_i \sigma^{i/n}$ for *some* coefficients a_i and integer n. We find this series by assuming that $s = c_0\sigma^{\gamma_0} + c_1\sigma^{\gamma_0+\gamma_1} + c_2\sigma^{\gamma_0+\gamma_1+\gamma_2} + \ldots$, and then finding c_i and γ_i one by one. The exponents γ_i are defined as certain slopes of the Newton polygon [Cas00] for $f_m(s, \sigma) = 0$. After a while we form a hypothesis about the general formula for c_i and prove it. See Appendix B for a more detailed description of the Newton-Puiseux algorithm and for the proofs of the following theorems.

Theorem 2. *Let σ be as in Eq. (4). The Puiseux series of the unique positive root s of f_m is*

$$s = \sum_{i=0}^{\infty} c_i \sigma^{(i-1)/2} = \sigma^{-1/2} + \frac{m-1}{2} - \frac{1}{8}(m^2-1)\sqrt{\sigma} + O(\sigma) \ , \tag{6}$$

where $c_i = (-1)^{i+1} \frac{(m-1)}{2^i i!} \frac{((i-1)(m+1))!!}{((i-1)(m-1))!!}$.

Denote by $s_{(i)} := \sum_{i=0}^{i-1} c_i \sigma^{(i-1)/2}$ the sum of the first i elements of this Puiseux series. In practice, it suffices to know the values $s_{(1)} = \sigma^{-1/2}$ and $s_{(2)} = \sigma^{-1/2} + (m-1)/2$ (see Sect. 4.1).

Knowing the Puiseux series of s we may substitute it into the communication function com of Eq. (3), deriving thus the Puiseux series for the communication. The following theorem follows.

Theorem 3. *Let $f : \{0, \ldots, w-1\}^{\chi} \to \{0,1\}^{\ell}$ be computable by a polynomial-size w-ary branching program P_f of length m. The leveled LHE scheme of Sect. 3 for f has communication*

$$\ell + 2\sqrt{(w-1)\chi mk\ell} + \frac{w-1}{2}\chi(m+1)k + O(\ell^{-1/2})$$

and rate

$$1 - 2\sqrt{\frac{(w-1)\chi mk}{\ell}} + \frac{w-1}{2} \cdot \frac{\chi(7m-1)k}{\ell} + O(\ell^{-3/2}) \ .$$

See App. D for a more detailed statement (with a more precise series expression) and a proof.

4.1 Efficient Algorithm for Finding Integer Approximation of Root

Next, we propose a simple binary search algorithm that finds the best integer approximation to the unique positive root s of Eq. (5) in $\approx \log_2 m$ steps. Clearly, in our application, an integer approximation is sufficient. Let σ be as defined in Eq. (4). We first show that for $s_{(1)} = \sigma^{-1/2}$ and $s_{(2)} = \sigma^{-1/2} + (m-1)/2$ as defined in Thm. 2, $f_m(s_{(1)}, \sigma)$ is negative and $f_m(s_{(2)}, \sigma)$ is positive. (This result is quite technical, and its proof is given in App. C.) Since $0 < s_{(1)} < s_{(2)}$, we know that the only positive root s of $f_m(x, \sigma)$ is in the interval $(s_{(1)}, s_{(2)})$ of length $(m-1)/2$. We can compute the integer approximation to s (that is sufficient for our purposes) by using binary search in this interval, see Alg. 1. The proof of the following lemma is given in App. C.

Lemma 1. *$f_m(s_{(1)}, \sigma) < 0$ and $f_m(s_{(2)}, \sigma) \geq 0$. Moreover, $f_1(s_{(2)}, \sigma) = 0$ and $f_m(s_{(2)}, \sigma) > 0$ for $m > 1$.*

Theorem 4. *Alg. 1 finds the best integer approximation of s in $\approx \log_2((m-1)/2) \approx \log_2 m$ steps. Its computational complexity is dominated by $\approx \log_2 m$ evaluations of f_m.*

$s_L \leftarrow \lfloor s_{(1)} \rfloor$; $s_H \leftarrow \lceil s_{(2)} \rceil$;
while $s_H > s_L + 1$ **do**
  $s_M \leftarrow \lfloor (s_L + s_H)/2 \rfloor$;
  if $f_m(s_M, \sigma) > 0$ **then** $s_H \leftarrow s_M$;
  else $s_L \leftarrow s_M$;
end
if $\mathsf{com}(\chi, w, m, s_L, k, \ell) < \mathsf{com}(\chi, w, m, s_H, k, \ell)$ **then return** $s \leftarrow s_L$;
else return $s \leftarrow s_H$;

Algorithm 1: Finding integer approximation to root s by using binary search

Proof. The algorithm finds the unique unit interval $[s^*, s^* + 1]$ in the original interval that contains the root. It does so by using binary search. The number of the steps is clearly (approximately) logarithmic in the length of the interval, $(m - 1)/2$. □

5 Near Optimal Rate $(n, 1)$-CPIR

Given our homomorphic encryption the of a near optimal rate $(n, 1)$-CPIR is straightforward. Following [Lip09], in an $(n, 1)$-CPIR protocol, we let the client first generate a new Damgård-Jurik public and secret key pair, and then send to the server the public key together with encryptions of every individual bit of the index x. The server represents her database $\boldsymbol{f}$ as a compact branching program P_f that computes the function f where $f(x) := f_x$, and then evaluates securely the client's query on top of it. The client obtains the encrypted source value, and then decrypts it. When using the new leveled LHE scheme, the computational and communication complexity is as per Thm. 1. Thus, the resulting CPIR protocol has both optimal rate (when using the parameters derived in Sect. 4), and (given the database is sufficiently redundant) sublinear-in-n computational complexity. On the other hand, if the database is not redundant, then the server just represents it as an w-ary tree of length m.

In most of the applications, $w = 2$, which is also often (but not always) the optimal case. The following corollary exemplifies this.

Corollary 1. *Assume that the DCR assumption [Pai99] is true. There exists a CPA-secure $(n, 1)$-CPIR protocol with communication $\ell + 1.72 \cdot \log_2 n \cdot \sqrt{k\ell} + 2(\log_5^2 n + \log_5 n)k + O(\ell^{-1/2})$ and rate $1 - 1.72 \cdot \log_2 n \cdot \sqrt{k/\ell} + 2(7 \log_5^2 n - \log_5 n) \cdot \frac{k}{\ell} + O(\ell^{-3/2})$.*

Proof. Follows from preceeding discussion and Thm. 3 by setting $m = \chi = \log_w n$, and considering the full w-ary decision tree. Thus, the $(n, 1)$-CPIR protocol has communication complexity $\ell + 2\frac{\sqrt{w-1}}{\log_2 w} \log_2 n \cdot \sqrt{k\ell} + O(1)$. Since w is an integer, the second coefficient in this series is minimized when $w = 5$. □

We provide an example with concrete parameters below. Consider the setting of $(n, 1)$-CPIR, where each database element is a movie, and a paying client wishes to obtain one movie without the server knowing which movie she wants to see. Assume that $k = 3072$, $\ell = 10^6 k$ (about 380 Megabytes), $m = \chi = 16$, and $w = 2$. (Thus $n = 65536$, which allows to select between more movies than any of the current commercial online stores offers.) Then Alg. 1 starts with interval (s_L, s_H), where $s_L = 1000$ (communication of 3170723518 bits) and $s_H = 1008$ (3170720770 bits). After 3 steps, the interval is (s_L, s_H), where $s_L = 1007$ (3170720767 bits) and $s_H = 1008$ (3170720770 bits), and thus the algorithm outputs $s_H = 1007$ as the desired integer approximation of the optimal length parameter s. This results in communication 3170720767. The achieved rate is 0.968865. If $\ell = 10^7 k$ (i.e., approximately 3.8 Gigabytes, which is a realistic size of a high definition movie), then the achieved rate is already 0.989969.

6 Communication complexity lower bounds in restricted models

In order to study the optimality of our constructions, we introduce two abstract models that capture the power of leveled homomorphic schemes that are constructed in a black-box fashion from an additive homomorphic public-key encryption (PKE). We provide lower-bounds on the communication complexity of private-information-retrieval in these models. We note that PIR be seen as an instance of large-output branching program evaluation and our results can be generalized to yield lower bounds for homomorphically evaluating branching programs.

We first derive a useful lemma in a restricted model where two-party protocols are implemented non-interactively via only access to a multivariate polynomial evaluation functionality. An important corollary of this is that when we use multivariate polynomial evaluation over degree 1 polynomials the best communication complexity we can obtain is $\Omega(\sqrt{n})$ where n is the cardinality of the client input domain.

The usefulness of the result will come by a further observation (see below) that additive homomorphic encryption that operates without "layering" i.e., encoding ciphertexts within plaintexts and rencrypting, can achieve at best a degree 1 multivariate polynomial evaluation.

In particular, the client and the receiver are communicating via a multivariate polynomial evaluation oracle $\mathsf{MP}(v, Q_1, \ldots, Q_t)$, operating over a finite field $\mathbb{F}$, where $v \in \mathbb{F}^r$ is some encoding of the user's input $x \in \{0,1\}^{\ell \cdot n}$ and $Q_d \in \mathbb{F}[y_1, \ldots, y_r]$ and returns to the user a vector in $\mathbb{F}^t$ whose d-th coordinate equals $u_d = Q_d(v)$. If the maximum degree of each Q_d is m then we write the oracle as $\mathsf{MP}^m(\cdot)$.

The length of the oracle communication for a protocol $\mathtt{B}$ in this model is measured in terms of the size of the input provided by the client to the $\mathsf{MP}(\cdot)$ oracle and the size of the response of the oracle to the client. In the appendix E.2 we prove a slightly more general version of the following lemma.

Lemma 2. *Let $m \in \mathbb{N}$. Consider a two-round protocol $\mathtt{B}$ in the $\mathsf{MP}^m(\cdot)$ model over a field $\mathbb{F}$ that realizes $(n,1)$-CPIR protocol for ℓ-bit strings. Assume the communication complexity of $\mathtt{B}$ is such that the client transmits r values to $\mathsf{MP}(\cdot)$ and receives t values from it. It holds that $\binom{r+m}{m} \cdot t \geq \ell \cdot n / \log |\mathbb{F}|$.*

If we count also the field size in the above description we have that the total communication of the protocol is $(t + r)$ encodings of values in $\mathbb{F}$. In the case that $m = 1$ the communication will be minimized provided that we choose $t, r \approx \sqrt{\ell n / \log |\mathbb{F}|}$. In the general case, if $m \ll r$ we have $(r+1)^m \cdot t \geq \ell n / \log |\mathbb{F}|$, the communication to the oracle will be minimized when we choose $r = (mn\ell / \log |\mathbb{F}|)^{\frac{1}{m+1}} - 1$ resulting in a total communication of $\Omega((mn\ell)^{\frac{1}{m+1}})$.

We recall that our LHE scheme is built on top of a $(w,1)$-CPIR protocol implemented over an additive homomorphic public-key encryption (PKE).

We next introduce the black-box PKE model for an additive homomorphic PKE scheme. In this model, we consider two-party two-round protocols $\mathtt{B}$ that *realize* a functionality $F(\cdot,\cdot)$ as follows. If the client has input x and the server input $(\mathbf{f}, \mathsf{s})$ the user receives $F(x, (\mathbf{f}, \mathsf{s})) = \mathsf{s}_{\mathbf{f}(x)}$. Specifically, in an $(n,1)$-CPIR protocol, $X = [n]$ and $F(x, \langle s_1, \ldots, s_n \rangle) = s_x$, with $\mathbf{f}$ being the identity function. In our model, the protocol is parameterized by m levels. Each level d utilizes a corresponding additive-homomorphic PKE oracle that encrypts in an associated group G_d. The encrypted messages may be homomorphically added via oracle queries and then passed to the next level $d+1$ to be processed further. Finally, the client receives an element from G_m.

Specifically, a black-box 2-round protocol $\mathtt{B}$ is comprised of three algorithms $\Pi = (C_1, S, C_2)$ with the following characteristics.[10] $\mathtt{B}$ is parameterized by m that corresponds to the "levels"

[10] In the context of LHE, recall that C_1 represents $\mathsf{Enc}^m(\cdot)$, S is $\mathsf{Eval}(\cdot)$ and C_2 is $\mathsf{Dec}(\cdot)$.

of the encryptions that it utilizes. With each level we associate an additive group. The d-th group $(G_d, +)$ requires b_d bits for the representation of its elements and the underlying PKE is suitable for encrypting elements of the groups G_d in a way that addition can be homomorphically achieved between encrypted elements. The exact choice of $G_1, \ldots, G_m$ is independent of our arguments and we make no further assumption about these groups beyond the fact that the basic operation can be efficiently implemented. Note that the groups are not necessarily distinct.

The first algorithm of $\mathtt{B}$ denoted by C_1 has input x and access to a set of oracles $\mathsf{Enc}^d(\cdot)$ for $d = 1, \ldots, m$ that operate as follows: $\mathsf{Enc}^d(\cdot)$ receives as input a plaintext m_0 so that $m \in G_d$; it returns a ciphertext ψ encrypting m_0. We say that ψ is a d-th level encryption of m_0.

C_1 using oracle access to $\{\mathsf{Enc}^d(\cdot)\}_d$ produces a sequence of ciphertexts $\psi_1, \ldots, \psi_t$ (that may belong to different levels). The algorithm S takes as input s as well as a sequence of ciphertexts $\langle \psi_1, \ldots, \psi_t \rangle$ (presumably the output of C_1). S has access to the oracle $\mathsf{Enc}^d(\cdot)$ and an oracle $\mathsf{Add}^d(\cdot)$ that takes as input a pair of ciphertexts (ψ_1, ψ_2) and provided that ψ_1, ψ_2 are d-level encodings encrypting the elements $m_1, m_2 \in G_d$, it returns a ciphertext ψ that encrypts the element $m_1 + m_2 \in G_d$. S terminates with a sequence of ciphertexts $\psi_1, \ldots, \psi_{t'}$. Finally, C_2 operates on input $x, \psi'_1, \ldots, \psi'_{t'}$ and returns an output z. Algorithm C_2 has oracle access to $\mathsf{Enc}^d(\cdot), \mathsf{Dec}^d(\cdot)$. The oracle $\mathsf{Dec}^d(\cdot)$ receives as input ψ and if ψ is a valid encryption of some $m_0 \in G_d$ it returns m_0.

Note that we assume perfect correctness from the underlying encryption, i.e., whenever ψ is an oracle answer to a query m_0 by $\mathsf{Enc}^d(\cdot)$, submitting ψ to $\mathsf{Dec}(\cdot)$ will return the value m_0.

Definition 3. *A protocol* $\mathtt{B}$ *realizes the functionality* $F(\cdot,\cdot)$ *in the black-box additive PKE model if it holds that for all* x, s *the output* z *of the* C_2 *algorithm in the interaction as defined above satisfies* $z = F(x, \mathsf{s})$ *for all correct instantiations of the* $\mathsf{Enc}^d(\cdot), \mathsf{Dec}^d(\cdot), \mathsf{Add}^d(\cdot)$ *oracles.*

An important feature of our approach is that the protocol $\mathtt{B}$ in the black-box PKE is completely agnostic in the implementation of the underlying PKE scheme. Thus by manipulating the PKE oracle one may deconstruct protocol $\mathtt{B}$ to simpler subprotocols for which we can analyze their communication complexity using the multivariate polynomial argument that was presented above. Specifically, for *canonical* protocols that satisfy certain simple requirements (see the appendixE for a full description) we prove the following.

Theorem 5. *Let* $\mathtt{B}_{\mathbf{f}}$ *be any canonical two-party protocol in the black-box additive PKE model that realizes* F *with* $F(x, (\mathbf{f}, \mathsf{s})) = s_{\mathbf{f}(x)}$ *for* $\mathbf{f} : X \to [n]$ *and* $\mathsf{s} \in (\{0,1\}^\ell)^n$. *Suppose that the total communication complexity of* $\mathtt{B}_{\mathbf{f}}$ *with respect to* n *is* $O_n(\log^\lambda n)$, *for some positive integer* λ; *then its total communication complexity is at least*

$$\ell + \Omega\left(\frac{\log n}{\log\log n}\right) \cdot \sqrt{\ell k} + \Omega\left(\frac{\log n}{\log\log n}\right) \cdot k \ .$$

Based on this we easily obtain the ceiling on the best possible communication rate for CPIR in the black-box additive PKE model which equals $1 - \Omega(\log n / \log\log n) \cdot \sqrt{k/\ell} + \Theta_\ell(\ell^{-1})$.

References

Bar86. David A. Mix Barrington. Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 1–5, Berkeley, California, USA, 28–30 May 1986. 3

BHR95. Yuri Breitbart, Harry B. Hunt III, and Daniel J. Rosenkrantz. On The Size of Binary Decision Diagrams Representing Boolean Functions. *Theoretical Computer Science*, 145(1&2):45–69, 1995. 2

Cas00. Eduardo Casas-Alvero. *Singularities of Plane Curves*, volume 276 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, September 2000. 1, 4, B, B

CFFC11. Ana Charpentier, Caroline Fontaine, Teddy Furon, and Ingemar Cox. An asymmetric fingerprinting scheme based on tardos codes. In *Proceedings of the 13th international conference on Information hiding*, IH'11, pages 43–58, Berlin, Heidelberg, 2011. Springer-Verlag. 1

CLO+13. Ashish Choudhury, Jake Loftus, Emmanuela Orsini, Arpita Patra, and Nigel P. Smart. Between a Rock and a Hard Place: Interpolating between MPC and FHE. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 221–240, Bangalore, India, December 1–5, 2013. Springer, Heidelberg. 1

CMS99. Christian Cachin, Silvio Micali, and Markus Stadler. Computational Private Information Retrieval with Polylogarithmic Communication. In Stern [Ste99], pages 402–414. 1

Cob66. Alan Cobham. The Recognition Problem for the Set of Perfect Squares. In *FOCS 1966*, pages 78–87, Berkeley, California, October 23–25, 1966. IEEE Computer Society. 2

DFH12. Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In *TCC*, pages 54–74, 2012. 1

DJ01. Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 119–136, Cheju Island, Korea, February 13–15, 2001. Springer, Heidelberg. 1, 2

DZ13. Ivan Damgård and Sarah Zakarias. Constant-overhead secure computation of boolean circuits using preprocessing. In *TCC*, pages 621–641, 2013. 1

Gen09a. Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, September 2009. 3, 3

Gen09b. Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In Michael Mitzenmacher, editor, *STOC 2009*, pages 169–178, Bethesda, Maryland, USA, May 31 — June 2, 2009. ACM Press. 1, 3

GHS12. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 850–867, 2012. 5

Gil91. George T. Gilbert. Positive Definite Matrices and Sylvester's Criterion. *The American Mathematical Monthly*, 98(1):44–46, 1991. A

GM82. Shafi Goldwasser and Silvio Micali. Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 365–377, San Francisco, California, USA, May 5–7, 1982. ACM. 1

GR05. Craig Gentry and Zulfikar Ramzan. Single-Database Private Information Retrieval with Constant Communication Rate. In Luis Caires, Guiseppe F. Italiano, Luis Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP 2005*, volume 3580 of *LNCS*, pages 803–815, Lisboa, Portugal, 2005. Springer, Heidelberg. 1

IKM+13. Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, Claudio Orlandi, and Anat Paskin-Cherniavsky. On the power of correlated randomness in secure computation. In *TCC*, pages 600–620, 2013. 1

IP07. Yuval Ishai and Anat Paskin. Evaluating Branching Programs on Encrypted Data. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 575–594, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg. 1, 3

KO97. Eyal Kushilevitz and Rafail Ostrovsky. Replication is Not Needed: Single Database, Computationally-Private Information Retrieval. In *FOCS 1997*, pages 364–373, Miami Beach, Florida, October 20–22, 1997. IEEE Computer Society. 1

Lip05. Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Communication. In Jianying Zhou and Javier Lopez, editors, *ISC 2005*, volume 3650 of *LNCS*, pages 314–328, Singapore, September 20–23, 2005. Springer, Heidelberg. 1, 8, 2

Lip09. Helger Lipmaa. First CPIR Protocol with Data-Dependent Computation. In Donghoon Lee and Seokhie Hong, editors, *ICISC 2009*, volume 5984 of *LNCS*, pages 193–210, Seoul, Korea, December 2–4, 2009. Springer, Heidelberg. 1, 8, 2, ii, 3, 9, 5

Lip11. Helger Lipmaa. Efficient multi-query cpir from ring-lwe. *IACR Cryptology ePrint Archive*, 2011:595, 2011. 6

NN01. Moni Naor and Kobbi Nissim. Communication preserving protocols for secure function evaluation. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *STOC*, pages 590–599. ACM, 2001. 1

Pai99. Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Stern [Ste99], pages 223–238. 1, 2, 1

Pfi96. Birgit Pfitzmann. Trials of traced traitors. In Ross J. Anderson, editor, *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 49–64. Springer, 1996. 1

PS96. Birgit Pfitzmann and Matthias Schunter. Asymmetric fingerprinting (extended abstract). In Ueli M. Maurer, editor, *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 1996. 1

PW97. Birgit Pfitzmann and Michael Waidner. Asymmetric fingerprinting for larger collusions. In Richard Graveman, Philippe A. Janson, Clifford Neumann, and Li Gong, editors, *ACM Conference on Computer and Communications Security*, pages 151–160. ACM, 1997. 1

Ste98. Julien P. Stern. A New And Efficient All Or Nothing Disclosure of Secrets Protocol. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT '98*, volume 1514 of *LNCS*, pages 357–371, Beijing, China, October 18–22, 1998. Springer, Heidelberg. 1

Ste99. Jacques Stern, editor. *EUROCRYPT 1999*, volume 1592 of *LNCS*, Prague, Czech Republic, May 2–6, 1999. Springer, Heidelberg. 6

SYY99. Tomas Sander, Adam Young, and Moti Yung. Non-Interactive CryptoComputing For NC^1. In *FOCS 1999*, pages 554–567, New York, NY, USA, 17–18 October 1999. IEEE Computer Society. 1

Wal13. Robert J. Walker. *Algebraic Curves*. Springer, October 4, 2013. B

Weg00. Ingo Wegener. *Branching Programs and Binary Decision Diagrams: Theory and Applications*. Monographs on Discrete Mathematics and Applications. Society for Industrial Mathematics, 2000. 2

A Derivation of Global Minimum

Here we give some details and explanations to Section 4.

Assuming that the rest of the parameters are fixed, the multivariate communication function com of Eq. (2) is minimized (or maximized) in $\boldsymbol{s} = (s_1, \ldots, s_m)$, when *the gradient* $\nabla\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell)$ is 0, i.e., $\partial\mathsf{com}/\partial s_1 = \cdots = \partial\mathsf{com}/\partial s_m = 0$, and the Hessian matrix

$$H(m, \boldsymbol{s}) := \begin{pmatrix} \frac{\partial^2\mathsf{com}}{\partial s_1 \partial s_1} & \frac{\partial^2\mathsf{com}}{\partial s_1 \partial s_2} & \cdots & \frac{\partial^2\mathsf{com}}{\partial s_1 \partial s_m} \\ \frac{\partial^2\mathsf{com}}{\partial s_2 \partial s_1} & \frac{\partial^2\mathsf{com}}{\partial s_2 \partial s_2} & \cdots & \frac{\partial^2\mathsf{com}}{\partial s_2 \partial s_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\mathsf{com}}{\partial s_m \partial s_1} & \frac{\partial^2\mathsf{com}}{\partial s_m \partial s_2} & \cdots & \frac{\partial^2\mathsf{com}}{\partial s_m \partial s_m} \end{pmatrix}$$

is positive definite. By the Sylvester's criterion a matrix M is positive definite iff the determinants associated with all upper-left submatrices of M are positive, [Gil91].

If s_j is such that $s_j \neq s_i^{\max}$ for any i, then the value of s_j is inconsequential for the communication complexity, and thus we do not have to optimize it. Thus, assume that $\zeta_j = |\{i : s_j = s_i^{\max}\}| \geq 1$. Then

$$\frac{\partial\mathsf{com}}{\partial s_j} = (w-1)\zeta_j k - \frac{\ell}{s_j^2} \cdot \frac{\prod_{i=1:i\neq j}^{m}(s_i+1)}{\prod_{i=1:i\neq j}^{m} s_i} = (w-1)\zeta_j k - \frac{\ell}{(s_j+1)s_j} \cdot \frac{\prod_{i=1}^{m}(s_i+1)}{\prod_{i=1}^{m} s_i} = 0$$

and thus,

$$\frac{(w-1)\zeta_j k}{\ell} \cdot \frac{\prod_{i=1}^{m} s_i}{\prod_{i=1}^{m}(s_i+1)} = \frac{1}{(s_j+1)s_j} \ .$$

for every j. In particular, this means that for every $i \neq j$, $s_i(s_i+1) = s_j(s_j+1)$. Thus either $s_i = s_j$ or $s_i = -1 - s_j$. But since s_i and s_j both have to be positive, we get that $s_1 = \cdots = s_m =: s$ for some s.

In the general case

$$\frac{\partial^2\mathsf{com}}{\partial s_i^2} = \frac{2\ell \prod_{\kappa\neq i}(s_\kappa+1)}{s_i^2 \prod_\kappa s_\kappa} \quad \text{and}$$
$$\frac{\partial^2\mathsf{com}}{\partial s_i \partial s_j} = \frac{\ell \prod_{\kappa\neq i,\kappa\neq j}(s_\kappa+1)}{s_i s_j \prod_\kappa s_\kappa} \quad \text{for } i \neq j \ .$$

Note that those values do not depend on m.

Let $\boldsymbol{s}^\star = (s^\star, \ldots, s^\star)$ be a solution of Eq. (2). All upper-left submatrices of $H(m, \boldsymbol{s})$ are circulant matrices with determinants of form

$$\begin{vmatrix} a & b & b & \vdots & b \\ b & a & b & \vdots & b \\ \ldots & \ldots & \ldots & \ddots & \ldots \\ b & b & b & \vdots & a \end{vmatrix} = b(a-b)^\kappa \left(\frac{1}{b} + \frac{\kappa}{a-b}\right) \ ,$$

where $\kappa \in \{0, \ldots, m-1\}$,

$$a = \frac{\partial^2 \mathsf{com}(\chi, w, m, \boldsymbol{s}^\star, k, \ell)}{\partial s_1^2} = \ldots = \frac{\partial^2 \mathsf{com}(\chi, w, m, \boldsymbol{s}^\star, k, \ell)}{\partial s_m^2} = \frac{2\ell(s^\star+1)^{\kappa-1}}{(s^\star)^{\kappa+2}} > 0$$

and

$$b = \frac{\partial^2 \mathsf{com}(\chi, w, m, \boldsymbol{s}^\star, k, \ell)}{\partial s_i \partial s_j} = \frac{\ell(s^\star+1)^{\kappa-2}}{(s^\star)^{\kappa+2}} > 0 \quad \text{for all} \quad i, j \in \{0, \ldots, \kappa-1\},$$

and $a - b > 0$. Thus, the Hessian matrices $H(m, \boldsymbol{s}^\star)$ are positive definite for all $m > 1$.

Any local minimum of a convex function is also a global minimum. A continuous, twice differentiable function of several variables is convex on a convex set if and only if its Hessian matrix is positive semidefinite on the interior of the convex set. Since a positive definite matrix is also positive semidefinite, $\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell)$ is a convex function on a convex domain $\boldsymbol{s} > 0$ and $\boldsymbol{s}^\star = (s^\star, \ldots, s^\star)$ is indeed a global minimum.

B Proof of Thm. 2

Proof. First, rewrite Eq.(5) as

$$f_m(s, \sigma) := \sum_{i,j=0}^{m+1} b_{ij} s^i \sigma^j = 0. \tag{7}$$

The solution $s(\sigma)$ of Eq. (7) will be constructed recursively as a Puiseux series in σ. Computing a power series expansion for $s(\sigma)$ can be seen as solving a polynomial equation in one variable over the field of Puiseux series. Since the field of formal Puiseux series is algebraically closed (see Puiseux's Theorem from [Wal13]), a root can always be found.

The root of function $f_m(s, \sigma)$ is of the form

$$s(\sigma) = c_0 \sigma^{\gamma_0} + c_1 \sigma^{\gamma_0+\gamma_1} + c_2 \sigma^{\gamma_0+\gamma_1+\gamma_2} + \ldots,$$

with $c_i \neq 0$, $\gamma_i \in \mathbb{Q}$, $\gamma_i > 0$ for all i, or, otherwise, it can be written as

$$s(\sigma) = \sigma^{\gamma_0}(c_0 + s_1) \ , \quad \text{where} \quad s_1 = c_1 \sigma^{\gamma_1} + c_2 \sigma^{\gamma_1+\gamma_2} + c_3 \sigma^{\gamma_1+\gamma_2+\gamma_3} \ldots \ .$$

$$s(\sigma) = \sigma^{\gamma_0}(c_0 + \sigma^{\gamma_1} s_2) \ , \quad \text{where} \quad s_2 = c_1 + c_2 \sigma^{\gamma_2} + c_3 \sigma^{\gamma_2+\gamma_3} \ldots \ ,$$

$$s(\sigma) = \sigma^{\gamma_0}(c_0 + \sigma^{\gamma_1}(c_1 + \sigma^{\gamma_2} s_3)) \ , \quad \text{where} \quad s_3 = c_2 + c_3 \sigma^{\gamma_3} + \ldots \ ,$$

and so on. According to the Newton-Puiseux algorithm [Cas00], the values of γ_i are defined as a certain slopes of Newton polygons of $f_m^{(i)}(s, \sigma)$, c_i are defined from letting terms of lowest order in equation $f_m^{(i)}(s, \sigma) = 0$ ($f_m^{(0)}(s, \sigma) := f_m(s, \sigma)$) be equal to zero. Newton polygon is the smallest convex polygon in the affine plane over $\mathbb{Q}$, which contains all the points $P_i = (i, j)$,

where (i,j) are the pairs of indices (i,j) of $f_m(s,\sigma) = \sum b_{ij} s^i \sigma^j$. Those faces of the Newton polygon, s.t. all the P_i's lie on or above the corresponding line, have possible values for γ_0 as their negative slopes. A detailed proof of the fact, that the Newton's Polygon Method can be performed on any polynomial f and that it actually yields Puiseux series is given in [Wal13].

Now we start from determining γ_0 and c_0. Since, $f_m(s,\sigma)$ has a non-zero b_{ij} iff

$(i,j) = (i,0)$ for $i \in [0, m-1]$, or
$(i,j) = (m+1, 1)$,

the Newton polygon method gives us that the lower convex hull of the non-zero points (i,j) has one non-horizontal slope, from $(m-1,0)$ to $(m+1,1)$. This slope can be written as $y = \frac{1}{2}x - \frac{m-1}{2} = -\gamma_0 x + \beta_0$, where $\gamma_0 = -\frac{1}{2}$ and $\beta_0 = -\frac{m-1}{2}$. Thus,

$$s(\sigma) = \sigma^{-1/2}(c_0 + s_1).$$

To find c_0 we substitute $\sigma^{-1/2}(c_0 + s_1)$ for s in $f_m(s,\sigma)$, getting

$$\begin{aligned}
f_m(s,\sigma) =& f_m\left(\sigma^{-1/2}(c_0+s_1),\sigma\right) = \sigma\left(\sigma^{-1/2}(c_0+s_1)\right)^{m+1} - \left(\sigma^{-1/2}(c_0+s_1)+1\right)^{m-1} \\
=& \sigma^{-(m-1)/2} \sum_{i=0}^{m+1} \binom{m+1}{i} c_0^{m+1-i} s_1^i - \sum_{i=0}^{m-1} \binom{m-1}{i} \sigma^{-i/2} \sum_{j=0}^{i} \binom{i}{j} c_0^{i-j} s_1^j \\
=& \sigma^{-(m-1)/2} \left(\sum_{i=0}^{m+1} \binom{m+1}{i} c_0^{m+1-i} s_1^i - \sum_{j=0}^{m-1} \binom{m-1}{j} c_0^{m-1-j} s_1^j \right) \\
& - \sum_{i=0}^{m-2} \binom{m-1}{i} \sigma^{-i/2} \sum_{j=0}^{i} \binom{i}{j} c_0^{i-j} s_1^j \ .
\end{aligned}$$

The terms of lowest order must cancel. We equate to zero the coefficient of $\sigma^{-(m-1)/2}$ getting $c_0^{m+1} - c_0^{m-1} = 0$, or $c_0^2 = 1$, or $c_0 \in \{-1, 0, 1\}$. Since we are interested only in positive non-trivial solutions, we skip $c_0 \in \{-1, 0\}$, and obtain $c_0 = 1$. Thus,

$$s(\sigma) = \sigma^{-1/2}(1 + c_1\sigma^{\gamma_1} + c_2\sigma^{\gamma_1+\gamma_2} + \dots) \ .$$

Substituting $c_0 = 1$ into $f_m(\sigma^{-1/2}(c_0 + s_1), \sigma)$ and multiplying the result with $\sigma^{-\beta_0} = \sigma^{(m-1)/2}$, we'll get a new polynomial

$$f_m^{(1)}(s_1, \sigma) = \sigma^{(m-1)/2} f_m(\sigma^{-1/2}(1 + s_1), \sigma).$$

from which next γ_2 and c_2 will be defined. To construct Newton polygon for $f_m^{(1)}(s_1, \sigma)$, we rewrite it as

$$\begin{aligned} f_m^{(1)}(s_1,\sigma) &:= \sum_{i=0}^{m+1} \binom{m+1}{i} s_1^i - \sum_{i=0}^{m-1} \binom{m-1}{i} \sigma^{(m-1-i)/2} (s_1+1)^i \\ &:= \left(s_1^{m+1} + (m+1)s_1^m + \sum_{i=0}^{m-1} \binom{m+1}{i} s_1^i \right) - \\ &\quad \left((s_1+1)^{m-1} + \sum_{i=0}^{m-2} \binom{m-1}{i} \sigma^{(m-1-i)/2} (s_1+1)^i \right) \\ &= s_1^{m+1} + (m+1)s_1^m + \sum_{i=1}^{m-1} \left(\binom{m+1}{i} - \binom{m-1}{i} \right) s_1^i - \\ &\quad \sum_{i=0}^{m-2} \binom{m-1}{i} \sigma^{(m-1-i)/2} \sum_{j=0}^{i} \binom{i}{j} s_1^j \ . \end{aligned}$$

Thus, the polynomial $f_m^{(1)}(s_1, \sigma) = \sum b_{ij} s_1^i \sigma^j$ has a non-zero coefficient b_{ij} iff

$(i, j) = (x, 0)$ for $x \in [1, m+1]$, or
$(i, j) = \left(x, \frac{m-1-y}{2}\right)$ for $x \in [0, y]$ and $y \in [0, m-2]$.

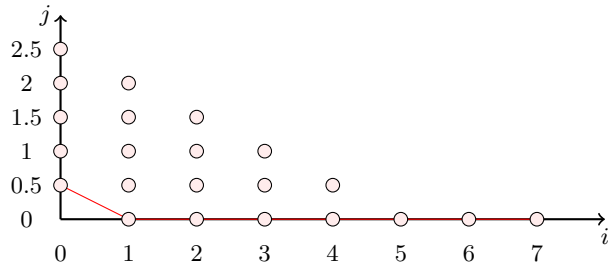

Fig. 4. Newton polygon for f_m^*, where $m = 6$

As seen from Fig. 4, the Newton polygon of $f_m^{(1)}(s_1, \sigma)$ consists of a single segment with only two vertices, one on each axis. That segment has equation $y = -\gamma_1 x + \beta_1 = -\frac{1}{2}x + \frac{1}{2}$. Such situation allows to compute the rest c_i by letting lowest powers of σ cancel. By the Newton-Puiseux algorithm [Cas00], from now on the powers of σ are going to jump by the denominator of γ_1; in other words $\gamma_1 = \gamma_2 = \gamma_3 = \cdots = 1/2$.

Now, we can calculate the rest of the c_i directly from the function obtained from substitution of s_1 in $f_m^{(1)}(s_1, \sigma)$ where

$$s_1 = c_1\sigma^{1/2} + c_2\sigma^{2/2} + c_3\sigma^{3/2} + \cdots = \sum_{i=1}^{\infty} c_i \sigma^{i/2}$$

and equating to zero all terms which contain σ of degree $\frac{n}{2}$, for all $n \geq 1$. For that we will consider the equality

$$f_m^{(1)}(s_1, \sigma) = \sigma^{\frac{m-1}{2}} f_m(\sigma^{-1/2}(s_1+1), \sigma) = (s_1+1)^{m+1} - (s_1 + 1 + \sigma^{1/2})^{m-1} \ .$$

Now, $s_1 + 1 = \sum_{i=0}^{\infty} c_i \sigma^{i/2}$ and $s_1 + 1 + \sigma^{1/2} = \sum_{i=0}^{\infty} c'_i \sigma^{i/2}$, where $c'_i = c_i$ for $i \neq 1$, and $c'_1 = c_1 + 1$.

If n is a natural number and i and a_0 are invertible, then

$$\left(\sum_{k=0}^{\infty} a_k X^k\right)^n = \sum_{k=0}^{\infty} b_k X^k, \qquad \text{where } b_0 = a_0^n, \text{ and } b_i = \frac{1}{i a_0} \cdot \sum_{k=1}^{i} (kn - i + k) a_k b_{i-k},\ i \geq 1.$$

Thus,

$$f^*_m(s_1, \sigma) = \sum_{i=0}^{\infty} \left(\delta_i - \delta'_i\right) \sigma^{i/2},$$

where $\delta_0 = \delta'_0 = 1$,

$$\delta_i = \frac{1}{i} \sum_{k=1}^{i} \left(k(m+1) - i + k\right) c_k \delta_{i-k}, \qquad \text{and} \qquad \delta'_i = \frac{1}{i} \sum_{k=1}^{i} \left(k(m-1) - i + k\right) c'_k \delta'_{i-k}\ , \quad (8)$$

for $i > 0$, since

$$(s_1 + 1)^{m+1} = \sum_{i=0}^{\infty} \delta_i \sigma^{i/2}, \qquad (s_1 + \sigma^{1/2} + 1)^{m-1} = \sum_{i=0}^{\infty} \delta'_i \sigma^{i/2}.$$

Collecting terms with $\sigma^{t/2}$, $t \geq 1$ in $f^*_m(s_1, \sigma)$, we get equations for calculating the rest of c_{t+1}. More precisely, we first collect terms that correspond $t = 1$, and then derive a recursive formula for the case $t > 2$. Via this recursive formula, we compute c_i for small i, and then guess the general value of c_i.

Consider first terms containing $\sigma^{1/2}$: $\delta_1 = \delta'_1$. According to Eq. (8),

$$\delta_1 - \delta'_1 = (m+1)c_1 - (m-1)(c_1 + 1) = 0\ ,$$

and thus

$$c_1 = \frac{m-1}{2} \quad \text{and} \quad \delta_1 = (m+1)c_1 = \frac{(m-1)(m+1)}{2}\ .$$

We assume recursively that $\delta_j = \delta'_j$ for all $j < i$. Then, assuming $i \geq 2$,

$$\begin{aligned}
0 =& i(\delta_i - \delta'_i) = \sum_{k=1}^{i} \big(k(m+1) - i + k\big) c_k \delta_{i-k} - \sum_{k=1}^{i} \big(k(m-1) - i + k\big) c'_k \delta'_{i-k} \\
=& \big((m+1) - i + 1\big) c_1 \delta_{i-1} + \sum_{k=2}^{i} \big(k(m+1) - i + k\big) c_k \delta_{i-k} \big((m-1) - i + 1\big) c'_1 \delta'_{i-1} - \\
& - \sum_{k=2}^{i} \big(k(m-1) - i + k\big) c'_k \delta'_{i-k} \\
=& (m + 2 - i) c_1 \delta_{i-1} - (m-i)(c_1 + 1)\delta_{i-1} + 2 \sum_{k=2}^{i} k c_k \delta_{i-k} \\
=& 2c_1 \delta_{i-1} - (m-i)\delta_{i-1} + 2 \sum_{k=2}^{i} k c_k \delta_{i-k} \\
=& 2ic_i + 2 \sum_{k=1}^{i-1} k c_k \delta_{i-k} - (m-i)\delta_{i-1}\ .
\end{aligned}$$

Thus,

$$c_i = -\frac{1}{i}\left((i\right.$$

On the other hand,

$$f_m(s_{(2)},\sigma) = k\chi(w-1)\left(\sigma^{-1/2}+\frac{m-1}{2}\right)^{m+1} - \ell m\left(\sigma^{-1/2}+\frac{m+1}{2}\right)^{m-1} \ ,$$

with

$$\begin{aligned} k\chi(w-1)\left(\sigma^{-1/2}+\frac{m-1}{2}\right)^{m+1} &= k\chi(w-1)\sum_{i=0}^{m+1}\binom{m+1}{i}\sigma^{-i/2}\left(\frac{m-1}{2}\right)^{m+1-i} \\ &\geq k\chi(w-1)\sum_{i=2}^{m+1}\binom{m+1}{i}\sigma^{-i/2}\left(\frac{m-1}{2}\right)^{m+1-i} \\ &= k\chi(w-1)\sum_{i=0}^{m-1}\binom{m+1}{i+2}\sigma^{-(i+2)/2}\left(\frac{m-1}{2}\right)^{m-1-i} \\ &= \ell m\sum_{i=0}^{m-1}\binom{m+1}{i+2}\sigma^{-i/2}\left(\frac{m-1}{2}\right)^{m-1-i} \ , \end{aligned}$$

and

$$\ell m\left(\sigma^{-1/2}+\frac{m+1}{2}\right)^{m-1} = \ell m\sum_{i=0}^{m-1}\binom{m-1}{i}\sigma^{-i/2}\left(\frac{m+1}{2}\right)^{m-1-i} \ .$$

Now,

$$\begin{aligned} f_m(s_{(2)},\sigma) &\geq \ell m\sum_{i=0}^{m-1}\binom{m+1}{i+2}\sigma^{-1/2}\left(\frac{m-1}{2}\right)^{m-1-i} - \ell m\sum_{i=0}^{m-1}\binom{m-1}{i}\sigma^{-i/2}\left(\frac{m+1}{2}\right)^{m-1-i} \\ &= \ell m\sum_{i=0}^{m-1}\binom{m-1}{i}\sigma^{-i/2}\cdot\left(\frac{m+1}{2}\right)^{m-1-i}(A_{mi}-1) \geq 0 \ , \end{aligned}$$

where

$$A_{mi} = \frac{\binom{m+1}{i+2}}{\binom{m-1}{i}}\left(\frac{m-1}{m+1}\right)^{m-1-i} = \frac{m(m+1)}{(i+1)(i+2)}\left(\frac{m-1}{m+1}\right)^{m-1-i} \geq 1$$

for all $0 \leq i \leq m-1$.

We prove that $A_{mi} \geq 1$ for $i \in [0, m-1]$ by induction on $m \geq i+1$ for every fixed i. For the induction base case $m = i+1$, note that $A_{m,m-1} = 1$. For the inductive step, assume that $A_{mi} \geq 1$ for some $m \geq i+1$. Then

$$\begin{aligned} A_{m+1,i} &= \frac{(m+1)(m+2)}{(i+1)(i+2)}\cdot\left(\frac{m}{m+2}\right)^{m-i} = \frac{m(m+1)}{(i+1)(i+2)}\cdot\left(\frac{m}{m+2}\right)^{m-1-i} \\ &\overset{(*)}{\geq} \frac{m(m+1)}{(i+1)(i+2)}\cdot\left(\frac{m-1}{m+1}\right)^{m-1-i} = A_{mi} \geq 1 \ , \end{aligned}$$

since for every $i \geq 0$, $g(m) = \left(\frac{m-1}{m+1}\right)^{m-1-i}$ is an increasing function in m for $m \geq i+1$, or equivalently, for $i \leq m-1$. (We note that in (*), equality holds only when $i = m-1$.) □

D Proof of Thm. 3

Proof. Recall that the communication function is

$$\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell) = (w-1)k\chi(s+1) + \ell\left(1+\frac{1}{s}\right)^m,$$

where

$$s = \sum_{i=0}^{\infty} c_i \sigma^{\frac{i-1}{2}},$$

c_i are defined as in Theorem 2.

We find $s^{-1} = \sum_{j=0}^{\infty} c'_j \sigma^{\frac{j+1}{2}}$ from the condition $ss^{-1} = 1$, obtaining

$$c'_i = (-1)^i \frac{(m-1)}{2^i \cdot i!} \frac{((i+1)m + (i-3))!!}{((i+1)(m-1))!!}.$$

In particular,

$$c'_0 = 1, \quad c'_1 = -\frac{m-1}{2}, \quad c'_2 = \frac{(m-1)(3m-1)}{2^2 \cdot 2!}, \quad c'_3 = -\frac{(m-1)2m(2m-1)}{2^2 \cdot 3!}$$

and so on. Then $1 + s^{-1} = \sum_{i=0}^{\infty} d_i \sigma^{i/2}$, where $d_0 = 1$, $d_k = c'_{k-1}$ for $k \geq 1$. Raising power series $1 + s^{-1}$ to m-th power, we obtain

$$\left(\sum_{i=0}^{\infty} d_i \sigma^{i/2}\right)^m = \sum_{i=0}^{\infty} u_i \sigma^{i/2}, \quad \text{where} \quad u_0 = 1, \quad u_p = \frac{1}{p}\sum_{t=1}^{p} (tm - p + t) d_t u_{p-t}.$$

In particular, $u_1 = m$, $u_2 = 0$, and so on. Then

$$\begin{aligned}
\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell) &= (w-1)k\chi\left(1 + \sigma^{-1/2} + c_1 + \sum_{i=1}^{\infty} c_{i+1}\sigma^{i/2}\right) + \ell\left(1 + \sum_{j=1}^{\infty} u_j \sigma^{j/2}\right) \\
&= \ell + 2\sqrt{(w-1)\chi mk\ell} + \frac{1}{2}\chi k(m+1)(w-1) \\
&+ \frac{1}{\sqrt{\ell}} \frac{((w-1)\chi k)^{3/2} c_2}{\sqrt{m}} + \frac{1}{\ell} \frac{((w-1)\chi k)^2 c_3}{m} + \sum_{i=3}^{\infty} C_i \sigma^{i/2},
\end{aligned}$$

where C_t are defined as $C_t = (w-1)k\chi c_{t+1} + \ell u_t$, $t \geq 3$. Thus

$$\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell) = \ell + 2\sqrt{(w-1)\chi mk\ell} + \frac{1}{2}\chi k(m+1)(w-1) + O(\ell^{-1/2}).$$

Rate is equal to $\ell\,\mathsf{com}^{-1}(\chi, w, m, \boldsymbol{s}, k, \ell)$. Let $\mathsf{com}^{-1}(\chi, w, m, \boldsymbol{s}, k, \ell) = \sum_{i=0}^{\infty} a_i \ell^{\frac{2-i}{2}}$, then from $\mathsf{com}(\chi, w, m, \boldsymbol{s}, k, \ell) \cdot \mathsf{com}^{-1}(\chi, w, m, \boldsymbol{s}, k, \ell) = 1$ we get $a_0 = \ldots = a_3 = 0$, $a_4 = 1$, $a_5 = -2\sqrt{(w-1)\chi mk}$, $a_6 = \frac{1}{2}(7m-1)(w-1)\chi mk$ and so on, thus

$$\begin{aligned}
\ell\mathsf{com}^{-1}(\chi, w, m, \boldsymbol{s}, k, \ell) &= \ell\left(\frac{1}{\ell} - \frac{2\sqrt{(w-1)\chi mk}}{\ell\sqrt{\ell}} + \frac{\chi k(7m-1)(w-1)}{2\ell^2} + O(\ell^{-5/2})\right) \\
&= 1 - \frac{2\sqrt{(w-1)\chi mk}}{\sqrt{\ell}} + \frac{\chi k(7m-1)(w-1)}{2\ell} + O(\ell^{-3/2}).
\end{aligned}$$

E Communication complexity lower bounds in the black-box additive PKE model

We introduce the black-box PKE model for an additive homomorphic public-key encryption (PKE) scheme $\langle \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec} \rangle$. The PKE scheme is assumed to operate over at least one additive group that is a subset of its plaintext space. In terms of ciphertext length for the PKE we assume that the ciphertext size that corresponds to a b-bit plaintext is $a(b)$. Note that $a(b)$ also depends on k, the security parameter of the scheme. The objective of this model is to provide the lower bound arguments from which the communication rate of any leveled homomorphic scheme is restricted if it is constructed in black-box fashion from an underlying black-box additive homomorphic PKE. However, our results are general and we state them for arbitrary 2-round two-party computation protocols.

Specifically, a black-box 2-round protocol $\mathtt{B}$ is comprised of three algorithms $\Pi = (C_1, S, C_2)$ with the following characteristics.[11] $\mathtt{B}$ is parameterized by m that corresponds to the "levels" of the encryptions that it utilizes. With each level we associate an additive group. The d-th group $(G_d, +)$ requires b_d bits for the representation of its elements and the underlying PKE is suitable for encrypting elements of the groups G_d in a way that addition can be homomorphically achieved between encrypted elements. The exact choice of $G_1, \ldots, G_m$ is independent of our arguments and we make no further assumption about these groups beyond the fact that the basic operation can be efficiently implemented. Note that the groups are not necessarily distinct.

The first algorithm of $\mathtt{B}$ denoted by C_1 has input x and access to a set of oracles $\mathsf{Enc}^d(\cdot)$ for $d = 1, \ldots, m$ that operate as follows: $\mathsf{Enc}^d(\cdot)$ receives as input a plaintext m_0 so that $m \in G_d$; it returns a ciphertext ψ encrypting m_0. We say that ψ is a d-th level encryption of m_0.

C_1 using oracle access to $\{\mathsf{Enc}^d(\cdot)\}_d$ produces a sequence of ciphertexts $\psi_1, \ldots, \psi_t$ (that may belong to different levels). The algorithm S takes as input s as well as a sequence of ciphertexts $\langle \psi_1, \ldots, \psi_t \rangle$ (presumably the output of C_1). S has access to the oracle $\mathsf{Enc}^d(\cdot)$ and an oracle $\mathsf{Add}^d(\cdot)$ that takes as input a pair of ciphertexts (ψ_1, ψ_2) and provided that ψ_1, ψ_2 are d-level encodings encrypting the elements $m_1, m_2 \in G_d$, it returns a ciphertext ψ that encrypts the element $m_1 + m_2 \in G_d$. S terminates with a sequence of ciphertexts $\psi_1, \ldots, \psi_{t'}$. Finally, C_2 operates on input $x, \psi'_1, \ldots, \psi'_{t'}$ and returns an output z. Algorithm C_2 has oracle access to $\mathsf{Enc}^d(\cdot), \mathsf{Dec}^d(\cdot)$. The oracle $\mathsf{Dec}^d(\cdot)$ receives as input ψ and if ψ is a valid encryption of some $m_0 \in G_d$ it returns m_0.

Note that we assume perfect correctness from the underlying encryption, i.e., whenever ψ is an oracle answer to a query m_0 by $\mathsf{Enc}^d(\cdot)$, submitting ψ to $\mathsf{Dec}(\cdot)$ will return the value m_0.

Definition 4. *A protocol* $\mathtt{B}$ *realizes the functionality* $F(\cdot, \cdot)$ *in the black-box additive PKE model if it holds that for all* x, s *the output* z *of the* C_2 *algorithm in the interaction as defined above satisfies* $z = F(x, \mathsf{s})$ *for all correct instantiations of the* $\mathsf{Enc}^d(\cdot), \mathsf{Dec}^d(\cdot), \mathsf{Add}^d(\cdot)$ *oracles.*

For instance in a $(n, 1)$-CPIR protocol, $\mathsf{s} \in \{0,1\}^{\ell \cdot n}$, $x \in [n]$ and $F(x, \langle s_1, \ldots, s_n \rangle) = s_x$. More generally, we consider the setting where the server input is a pair $(\mathbf{f}, \mathsf{s})$ such that $\mathbf{f} : X \to [n]$ is a surjection and $\mathsf{s} \in \{0,1\}^{\ell \cdot n}$, the client input is a value $x \in X$ and $F(x, (\mathbf{f}, \mathsf{s})) = s_{\mathbf{f}(x)}$ (the case of CPIR is a special case by setting $X = [n]$ and $\mathbf{f}$ the identity function).

Now suppose that we have a specific additive homomorphic encryption scheme $\mathtt{PKE}$ that can be used to implement the oracles $\mathsf{Enc}^d(\cdot), \mathsf{Dec}^d(\cdot)$. It can be easily observed that the total communication of the protocol by $\mathtt{B}^{\mathtt{PKE}}$ is equal to $\mathsf{len}(pk) + \sum_{i=1}^{t} b_{l(i)} + \sum_{i=1}^{t'} b_{l'(i)}$ where $l(i)$ is the level of ciphertext ψ_i in the output of C_1 and $l'(i)$ is the level of ciphertext ψ'_i in the output of S; $\mathsf{len}(pk)$ is the length of the public-key.

[11] In the context of LHE, recall that C_1 represents $\mathsf{Enc}^m(\cdot)$, S is $\mathsf{Eval}(\cdot)$ and C_2 is $\mathsf{Dec}(\cdot)$.

The protocol $\mathtt{B}$ will be called a *canonical* protocol of depth m provided the following hold true:

- *Client Encoding Requirement.* The server will have available at least one encryption depending solely on client data for each level, i.e., at least one ciphertext with plaintext related to client input x in each of $G_1, \ldots, G_m$. Nevertheless, not all such ciphertexts will have to be produced directly and communicated by C_1. Specifically, C_1 using x produces some arbitrary encipherment of x from which all necessary ciphertexts of total length L can be generated by the server in some predetermined fashion.
- *Server Data Flow Requirement.* The way the server S utilizes its oracles obeys to the following rules: any input provided to the oracles $\mathsf{Enc}^1(\cdot), \mathsf{Add}^1(\cdot)$ may arbitrarily depend on $(\mathbf{f}, \mathsf{s})$ and constant values. In a similar vein, any input provided to the oracles $\mathsf{Enc}^d(\cdot), \mathsf{Add}^d(\cdot)$ for $d > 1$ may depend on $(d-1)$-level encryptions, $\mathbf{f}$ and constant values. Finally, the output of S consists solely of m-level encodings.

Some remarks are in place about the above abstraction and its relation to actual two-party protocols based on additive homomorphic encryption. The cost of accessing the oracles for both parties is assumed a single computational step. Moreover, we make no assumption about the underlying PKE beyond perfect correctness. We note that this assumption is made for convenience and our lower bound can be easily adjusted for the case that correctness is assumed to hold with overwhelming probability. Regarding the client encoding and server data flow requirement there seems to be nothing to be gained by violating these requirements. Specifically, in case the client does not provide an encoding for a certain level this means that the server's computation cannot interact with the client's private input at that level in any way except through re-encoding encodings of lower levels into that encoding level - a redundant operation.

E.1 First communication lower bound in the black-box additive PKE model

We establish a communication lower bound for a canonical protocol of depth m in the black-box additive PKE model. The following simple observation is at the core of the lower bound argument.

Lemma 3. *Assume an additive homomorphic* $\mathtt{PKE}$ *with ciphertext length* $a(b) \geq b + k$ *where* k *is the security parameter and a canonical black-box construction* $\mathtt{B}$ *of depth* m *that realizes* F*. Then, the protocol* $\mathtt{B}^{\mathtt{PKE}}$ *realizes* F *with communication complexity at least*

$$\ell \cdot \prod_{d=1}^{m} (1 + \frac{k}{b_d}) + mk + \sum_{d=1}^{m} b_d$$

where b_d *is the length needed to encode level-d plaintexts.*

Proof. In order to ensure correctness observe that ℓ output bits need to be transferred to the receiver. This follows from the fact that $F(x, \cdot)$ is a surjection over $\{0,1\}^\ell$ for any client input x.

At level 1, these bits will be encoded somehow and will require at least $\frac{\ell}{b_1} a(b_1)$ bits. Continuing iteratively it is easy to see that the initial ℓ plaintext bits will require at least $\ell \prod_{d=1}^{m} a(b_d)/b_d$ bits in order to be encoded at the m-level. Now given that $a(w) \geq w + k$ it follows that

$$\ell \prod_{d=1}^{m} a(b_d)/b_d \geq \ell \prod_{d=1}^{m} (1 + \frac{k}{b_d})$$

At the same time the receiver needs to transmit a ciphertext at each level; this means that it will need to send a total of ciphertext bits equal to

$$\sum_{d=1}^{m} a(b_d) \geq mk + \sum_{d=1}^{m} b_d$$

Combining the above we obtain the proof of the statement. □

Next we prove the following:

Theorem 6. *Assume an additive homomorphic* PKE *with ciphertext length $a(b) \geq b + k$ where k is the security parameter and a canonical black-box construction* B *of depth m that realizes F. Then, the protocol* $\mathtt{B}^{\mathtt{PKE}}$ *has communication complexity at least $\ell + 2m\sqrt{\ell k} + mk$ where k is the security parameter of the PKE.*

Proof. For the first case, set $m = 1$ in the lower bound of lemma 3. We obtain the $\ell(1 + k/w) + k + w = \ell + k + \ell k/w + w$. For fixed integers ℓ, k this function minimizes on the point $w = \sqrt{k\ell}$ from which the result follows.

For the general case recall that we need to minimize the expression

$$T := \ell \cdot \prod_{d=1}^{m} (1 + \frac{k}{b_d}) + mk + \sum_{d=1}^{m} b_d$$

for $b_i \geq 1$. Setting the partial derivatives of each b_d to zero we obtain that for each $d \in [m]$

$$b_d^2 = k\ell \cdot \prod_{d \neq j} (1 + \frac{k}{b_j}).$$

Multiplying by $(1 + k/b_d)$ we obtain

$$b_d^2 + kb_d = k\ell \cdot \prod_{j} (1 + \frac{k}{b_j}).$$

It follows that $b_d^2 + kb_d$ is constant. In particular, for any $i, j \in [m]$,

$$b_i^2 - b_j^2 + kb_i - kb_j = 0 \implies (b_i - b_j)(b_i + b_j + k) = 0 \implies b_i = b_j.$$

Observe now that

$$T \geq \ell(1 + \frac{k}{b})^m + mk + mb \geq \ell + m(b + \frac{k\ell}{b}) + mk$$

Using similar arguments as in the $m = 1$ case the result follows. □

E.2 A communication lower bound via multivariate polynomial evaluation.

An interesting communication lower bound can be derived by looking at two-party protocols that are implemented via multivariate polynomial evaluation. An important corollary of the results of this section is that when we use multivariate polynomial evaluation over degree 1 polynomials (which is what additive homomorphic encryption allows) the best communication complexity we can obtain is $\Omega(\sqrt{n})$ where n is the cardinality of the client input domain.

The model in this section considers two-round protocols $\mathtt{B}$ in an information theoretic setting where client and receiver (C_1, S, C_2) have access to a multivariate polynomial evaluation oracle $\mathsf{MP}(\cdot)$: specifically, for a given finite field $\mathbb{F}$ the oracle operates as $\mathsf{MP}(v, Q_1, \ldots, Q_t)$ where $v \in \mathbb{F}^r$ and $Q_d \in \mathbb{F}[x_1, \ldots, x_r]$ and returns to the user a vector in $\mathbb{F}^t$ whose d-th coordinate equals $u_d = Q_d(v)$. If the maximum degree of each Q_d is m then we write the oracle as $\mathsf{MP}^m(\cdot)$.

The communication complexity of the protocol $\mathtt{B}$ in this model is measured in terms of the size of the input provided by the client to the $\mathsf{MP}(\cdot)$ oracle and the size of the response of the oracle to the client.

Lemma 4. *Let $m \in \mathbb{N}$. Consider a two-round protocol $\mathtt{B}$ in the $\mathsf{MP}^m(\cdot)$ model over a field $\mathbb{F}$ that realizes F. Assume the communication complexity of $\mathtt{B}$ is such that the client transmits r values to $\mathsf{MP}(\cdot)$ and receives t values from it. It holds that $\binom{r+m}{m} \cdot t \geq \ell \cdot |X| / \log |\mathbb{F}|$.*

Proof. We fix a sequence of coins and we analyze the behavior of $\mathtt{B}$ deterministically over this coin selection. First we observe that C_1 determines a bijection $f : X \to \{v_1, \ldots, v_n\}$ with $n = |X|$ that maps the input of the client to a specific vector over $\mathbb{F}^r$; r is some arbitrary parameter $r \geq 1$ of the protocol corresponding to the input the client C_1 provides to the $\mathsf{MP}^m(\cdot)$ oracle and $\mathbb{F}$ is the underlying finite field.

The client protocol C_2 receives from the oracle a response in $\mathbb{F}^t$ and computes the appropriate value in $\{0,1\}^\ell$. C_2 determines a family of functions $\{g_i\}_{i=1,\ldots,n}$ with $g_i : \mathbb{F}^t \to \{0,1\}^\ell$.

We define now a function G that has domain $\{0,1\}^{n\ell}$ and maps a vector s to a subset of matrices $\mathbb{F}^{n \times t}$ so that $U \in G(\mathsf{s})$ if and only if $g_i(u_i) = F(x_i, \mathsf{s})$ for $i = 1, \ldots, n$ where u_i is the i-th row of U.

Given the above functions, we can now describe the operation of protocol $\mathtt{B}$ as follows. The client first computes the vector $f(x)$ and sends the r encodings of the elements of $\mathbb{F}$ that comprise $f(x)$ to $\mathsf{MP}(\cdot)$.

The server now will have to prepare the suitable polynomials to supply to $\mathsf{MP}(\cdot)$. Given the server input s, the server will define t polynomials $Q_1(\cdot), \ldots, Q_t(\cdot)$. We observe that in order for correctness to be satisfied it should be that $g_x(\langle Q_1(v_x), \ldots Q_t(v_x)\rangle) = F(x, \mathsf{s})$ for all $x \in X$.

It follows that the server will have to find a $U \in G(\mathsf{s})$ so that the t systems defined by the equations $\langle Q_1(v_i), \ldots, Q_t(v_i)\rangle = u_i$ for $i = 1, \ldots, n$ where u_i is the i-th row of U are solvable (if no such U exists the server necessarily has to fail). We remark that the way that the server finds the suitable U is of no importance in the argument.

Consider that due to the restriction in the $\mathsf{MP}^m(\cdot)$ oracle we want the polynomials Q_d to have degree m. Recall that $u_1, \ldots, u_n \in \mathbb{F}^t$ and that the vector of polynomials $\langle Q_1, \ldots, Q_t\rangle$ should match u_i when evaluated on v_i for all $i = 1, \ldots, n$. Given the restriction of the degree we have that a necessary condition for solvability would be that the values of each u_i when projected to a single coordinate should lie on a curve degree m. Given that there are $|\mathbb{F}|^{\binom{r+m}{m}}$ distinct polynomials over $\mathbb{F}$, the set of "solvable" matrices has cardinality $|\mathbb{F}|^{\binom{r+m}{m}t}$. We observe next that for any server input s, the set $G(\mathsf{s})$ should contain at least one solvable matrix U, thus it should hold that $|\mathbb{F}^{\binom{r+m}{m}t}| \geq |\{0,1\}^{\ell|X|}|$. Based on this we obtain that $\binom{r+m}{m} t \log |\mathbb{F}| \geq \ell |X|$ from which the result follows. □

If we count also the field size in the above description we have that the total communication of the protocol is $(t + r)$ encodings of values in $\mathbb{F}$. In the case that $m = 1$ the communication will be minimized provided that we choose $t, r \approx \sqrt{\ell n / \log |\mathbb{F}|}$. In the general case, if $m \ll r$ we have $(r+1)^m \cdot t \geq \ell n / \log |\mathbb{F}|$, the communication to the oracle will be minimized when we choose $r = (mn\ell / \log |\mathbb{F}|)^{\frac{1}{m+1}} - 1$ resulting in a total communication of $\Omega((mn\ell)^{\frac{1}{m+1}})$.

E.3 Second communication lower bound in the black-box additive PKE model

In this section we provide a lower bound for two-party protocols in the black-box PKE model that takes into account the size n of the database

Theorem 7. *Let $\mathtt{B}_{\mathbf{f}}$ be any canonical two-party protocol in the black-box additive PKE model that has depth m and realizes F with $F(x,(\mathbf{f},\mathsf{s})) = s_{\mathbf{f}(x)}$ for $\mathbf{f} : X \to [n]$ and $\mathsf{s} \in (\{0,1\}^\ell)^n$. Then, for any $\mathbf{f}$, it holds that the total communication complexity of $\mathtt{B}_{\mathbf{f}}$ is at least*

$$\frac{1}{2}(m+1)\left(n \cdot \frac{k+\ell+m}{m+1}\right)^{\frac{1}{m+1}}.$$

Proof. For some $\mathbf{f}$ we partially fix the server input to $(\mathbf{f},\cdot)$. In the resulting protocol $\mathtt{B}_{\mathbf{f}}$ the server provides only $\mathsf{s} \in (\{0,1\}^\ell)^n$ (recall the client provides $x \in X$ so that he obtains $\mathsf{s}_{\mathbf{f}(x)}$). Furthermore, since we are interested in the worst-case communication complexity of the protocol, we fix any source of randomness that the client or the server may be using.

Using $\mathtt{B}$ observe we can easily derive a depth-1 protocol for f. We execute (C_1, S, C_2) up to after the server completes the last call to an $\mathsf{Enc}_1(\cdot), \mathsf{Add}_1(\cdot)$ oracle, we collect all level-1 ciphertexts $c_1, \ldots, c_q$ and transmit them to the client. The client now continues the simulation of the server S as well as C_2 and produces the output. This is possible due to the server data flow requirement. The correctness of this protocol, call it $\mathtt{B}_1$, is immediate.

Now, for any subset $M \subseteq [q]$, consider a protocol $\mathtt{B}_1[M]$, that works as $\mathtt{B}_1$ with the difference that the client receives only the ciphertexts in $\{c_j : j \in M\}$ and the rest—in order to continue the simulation—he generates at random. Let M^* denote a subset of $[q]$ of minimum size, with the property that $\mathtt{B}_1[M^*]$ is always correct. Such a subset exists since $\mathtt{B}_1[[q]] = \mathtt{B}_1$, which is correct. If the client communicates r_1 elements in the first level and $|M^*| = n_1$, then Lemma 4 gives

$$(r_1+1)n_1 \geq n\ell/\log|G_1|.$$

Furthermore, we claim that for every ciphertext in $\{c_j : j \in M\}$ there is an execution where it is queried in Dec^1 oracle. This is because otherwise such a ciphertext could also be replaced by a random one contradicting the minimality of M^*. This implies that the protocol $\mathtt{B}_{2,\ldots,m}$ restricted in rounds $2, \ldots, m$ is a protocol for obtaining one out of n_1 words of length ψ_1 (the length of the ciphertexts at level 1).

Applying the above recursively and taking into account that ψ_d is the ciphertext length at level d we obtain that, for each level $d \in [m]$,

$$(r_d+1)n_d \geq n_{d-1}\psi_{d-1}/\log|G_d|,$$

with $n_0 = n$, $\psi_0 = l, G_0 = \{0,1\}^\ell$. From this and the fact that for each $d \in [m]$, $\psi_d \geq \log|G_d| + k$, we have that

$$2\psi_d r_d \geq (r_d+1)\psi_d \geq \frac{n_{d-1}}{n_d} \cdot \log(2^k|G_{d-1}|). \tag{11}$$

We now bound the communication cost T as follows

$$\begin{aligned}
T &\geq \sum_{d=1}^{m} \psi_d r_d + \psi_m n_m \\
&\geq (m+1)\left(2^{-m} n_0 \psi_m \prod_{d=1}^{m} \log(2^k |G_{d-1}|)\right)^{\frac{1}{m+1}} && \text{(AM-GM inequality and (11))} \\
&\geq (m+1)\left(2^{-m} n \log\left(2^k \sum_{d=0}^{m} |G_d|\right)\right)^{\frac{1}{m+1}} && (n_0 = n \text{ and } \psi_m \geq \log|G_m| + k) \\
&\geq \frac{1}{2}\cdot(m+1)\left(n \log\left(2^{k+m}|G_0|\right)^{\frac{1}{m+1}}\right)^{\frac{1}{m+1}} && \text{(AM-GM inequality inside the log)} \\
&\geq \frac{1}{2}\cdot(m+1)\left(n \cdot \frac{m+k+\ell}{m+1}\right)^{\frac{1}{m+1}}.
\end{aligned}$$

□

We have obtained a bound that if focused on the relationship of the number of elements n in the database and the number of levels m that a canonical protocol employs. Recall also the bound in Theorem 6, that is focused on the legth on each database entry ℓ. Combining these two we can state the following bound as their corollary.

Corollary 2 (Lower-bound for Private-Information-Retrieval). *Let $\mathtt{B}_{\mathbf{f}}$ be any canonical two-party protocol in the black-box additive PKE model that realizes F with $F(x,(\mathbf{f},\mathsf{s})) = s_{\mathbf{f}(x)}$ for $\mathbf{f} : X \to [n]$ and $\mathsf{s} \in (\{0,1\}^\ell)^n$. Suppose that the total communication complexity of $\mathtt{B}_{\mathbf{f}}$ with respect to n is $O_n(\log^\lambda n)$, for some positive integer λ; then its total communication complexity is at least*

$$\ell + \sqrt{\ell k}\cdot\Omega\left(\frac{\log n}{\log\log n}\right) + k\cdot\Omega\left(\frac{\log n}{\log\log n}\right).$$

Proof. By Theorem 7 we have $n^{\frac{1}{m+1}} = O_n(\log^\lambda n)$. It follows (by taking logarithms) that $m = \Omega\left(\frac{\log n}{\log\log n}\right)$. The bound then is a consequence of Theorem 6. □