Raphael R. Toledo, George Danezis, and Ian Goldberg

Lower-Cost ϵ-Private Information Retrieval

Abstract: Private Information Retrieval (PIR), despite being well studied, is computationally costly and arduous to scale. We explore lower-cost relaxations of information-theoretic PIR, based on dummy queries, sparse vectors, and compositions with an anonymity system. We prove the security of each scheme using a flexible differentially private definition for private queries that can capture notions of imperfect privacy. We show that basic schemes are weak, but some of them can be made arbitrarily safe by composing them with large anonymity systems.

Keywords: Private Information Retrieval, Anonymous communications, Private Queries, Differential Privacy

DOI 10.1515/popets-2016-0035
Received 2016-02-29; revised 2016-06-02; accepted 2016-06-02.

1 Introduction

Despite many years of research and significant advances, Private Information Retrieval (PIR) still suffers from scalability issues that were identified some time ago [31]: both information theoretic [10] (IT-PIR) and computational [26] PIR schemes require database servers to operate on all records for each private query to conceal the sought record. Thus, as the database grows, the time to process each query grows linearly in theory, and super-linearly in practice: as the data processed exceeds the CPU cache, it has to be fetched from the main memory and eventually persistent storage. Furthermore, in IT-PIR each query is processed by multiple PIR servers. As the number of servers increases, so do the communication and computation costs.

Yet the need to privately access large public databases is pressing: for example Certificate Transparency [22], which strengthens TLS certificate issuing, requires clients to look up certificates for sites they have visited, resulting in privacy concerns. The current size of the certificate database precludes trivial downloading of the entire set and requires PIR [23], but it cannot scale to the ubiquitous use of TLS in the future. More scalable systems are therefore needed, even at the cost of lowering the quality of protection[1]. Similarly, as the Tor network [13] grows it becomes untenable to broadcast information about all servers to all clients, and a private querying mechanism [24] will have to be implemented to prevent attacks based on partial knowledge of the network [11].

To address this challenge, we present designs that lower the traditional PIR costs, but leak more information to adversaries. Quantifying that leakage is therefore necessary and we propose a game-based differential privacy definition [14] to evaluate and minimize the risk introduced. This definition can also be used to demonstrate the inadequacy of folk designs for private queries: in the first design, a client queries an untrusted database by looking up the sought record along with other 'dummy' records [5, 19, 21] to confuse the adversary; in the second design, a client fetches a record from an untrusted database through an anonymity system [13, 15] to hide the correspondence between the client and the server.

The contributions of this paper are:

- We present and motivate a flexible differential privacy-based PIR definition, through a simple adversary distinguishability game, to analyze lighter-weight protocols and estimate their risk. This is necessary to quantify leakage, but can also capture systems that are arbitrarily private, including computationally and unconditionally private schemes.
- We argue that simple private retrieval systems using dummies and anonymous channels are not secure under this definition. A number of proposed systems have made use of such private query mechanisms, and we show they can lead to catastrophic loss of privacy when the adversary has side information.
- We present a number of variants of PIR schemes that satisfy our definition, and compare their security and cost. Our key result is that their composition with an anonymity system can achieve arbitrarily good levels of privacy, leading to highly secure alternatives to traditional PIR schemes.
- As a means to achieving the above we present a generic composition theorem of differentially private systems with anonymity channels, which is of independent interest.

Raphael R. Toledo: University College London (r.toledo@cs.ucl.ac.uk)
George Danezis: University College London (g.danezis@ucl.ac.uk)
Ian Goldberg: University of Waterloo (iang@cs.uwaterloo.ca)

1 The privacy offered today by Certificate Transparency is simply to have clients download a range of certificates instead of just one. See below for our analysis of this naive dummy requests mechanism.

– We present an optimization to reduce PIR costs and speed up an IT-PIR scheme by contacting fewer databases, and evaluate it using a Chor-like scheme as an example.

The rest of the paper is organized as follows. We present related work on PIR, anonymity systems, and the uses of differential privacy in the remainder of this section. After introducing the paper's notations, we define the threat model and present the privacy definitions in Section 2. We then demonstrate why the use of dummies and anonymity system alone does not guarantee privacy under our definitions in Section 3 and present our ϵ-private designs and an optimization to cut the computation cost in Sections 4 and 5. Finally in Section 6, we discuss the costs and efficiency of the designs before concluding the paper in Section 7.

1.1 Related Work

Private Information Retrieval (PIR) was introduced in 1995 by Chor et al. [10] to enable private access to public database records. As initially proposed, PIR provided information-theoretic privacy protection (IT-PIR) but required multiple non-collaborating servers each with a copy of the database. Later, Computational PIR (CPIR) [26] was proposed using a single server, but its practicality has been questioned as being slower than simply downloading the entire database at typical network speeds [31]. Since that time, however, newer CPIR schemes that are significantly faster than downloading the entire database have been proposed [1, 2, 25]. While IT-PIR offers perfect privacy—the confidentiality of the query cannot be breached even with unlimited resources—it is still a computational burden on multiple databases, since all records must be processed for each query and by each server. IT-PIR has been gradually enhanced over time with new capabilities, such as batch processing [20], multi-block queries [17] and tolerance to misbehaving servers [6]. Alternative approaches to scaling PIR include using trusted hardware elements [4].

Research on Anonymity Systems (AS) began in 1981 by David Chaum introducing the mixnet for anonymous e-mail [9]. Other AS applications were then studied, such as peer-to-peer and web browsing, in particular in the context of Onion Routing and Tor [13]. The Anonymity System accepts a batch of messages and routes them while hiding the correspondence between senders and receivers. Low-latency anonymity systems, however, may still fail under attacks such as traffic analysis and fingerprinting [32]. Cascade mix networks offer perfect privacy at the cost of lower performance [7]. In this work we will always consider an ideal anonymity system that can be abstracted, from the perspective of an adversary, as an unknown random permutation of input messages to output messages. Real-world instantiations are imperfect and security parameters may have to be adapted, or in the case of onion routing systems [13] some additional batching and mixing may be required.

Differential Privacy definitions and mechanisms were first presented in 2006 [14] to enable safe interactions with statistical databases. However, this definition has since been used in machine learning [30], cloud computing [27], and location indistinguishability together with PIR [3] to evaluate and minimize the privacy risk. Differentially private definitions have several advantages: they capture the intrinsic loss of privacy due to using a mechanism, and they are not affected by side information the adversary may hold. Well-known composition theorems can be applied. We note that Chor et al. [10] also make passing allusion to statistical and leaky definitions of PIR in their seminal paper, only to focus on perfectly information-theoretic schemes.

2 Definitions and ϵ-Privacy

In this work we characterize as PIR any system where a user inputs a secret index of a record in a public database, and eventually is provided with this record, without a defined adversary learning the index. We note that the systems we propose use different mechanisms from traditional IT-PIR and CPIR, and make different security assumptions. Yet they provide the same functionality and interface, and in many cases can be used as drop-in replacements for traditional PIR.

2.1 Notation

Entities. All systems we explore allow u users $\mathcal{U},\ i \in [\![1, u]\!]$, to perform q queries by sending p requests to the database system $\mathcal{DS}$. A database system $\mathcal{DS}$ is composed of $d \in \mathbb{N}$ replicated databases $\mathcal{DB}_{i \in [\![1,d]\!]}$. Each of them stores the same n records of standardized size b bits. We assume a cascade mix network provides an anonymous channel all users can access. We abstract this Anonymity System as one secure subsystem providing a perfectly secret bi-directional permutation between input and output messages.

When presenting mechanisms not using the anonymity system we will simply present the interactions of a single user with the database servers, and assume that all user queries can be answered by trivial parallel composition. However, when reasoning about PIR systems using an anonymity system, all user queries are assumed to transit though the anonymous channel.

Costs. This work studies PIR scalability, and we focus on analyzing costs on the server side, which is the performance bottleneck of current techniques. We denote the communication costs by C_c for the number of *bits* sent by the client to $\mathcal{DS}$, and by C_s for the number of *b-bit record blocks* sent by $\mathcal{DS}$ to the client. The computation cost C_p corresponds to the sum, for each record accessed, of the record access cost and the processing cost by the servers (e.g., the number of XORs), $C_p = N_{record\ access} \cdot (c_{acc} + c_{prc})$.

2.2 Privacy Definition

Threat Model. We consider an adversary $\mathcal{A}$ has corrupted d_a databases out of d, in order to discover the queries of a target user $\mathcal{U}_t$. These corrupted servers passively record and share all requests they observe to achieve this objective. We also assume that the adversary observes all the user's incoming and outgoing communication. However, in all presented systems, the requests are encrypted with the servers' public keys, and we assume that for communication with honest servers, only message timing, volume and size are ultimately visible to the adversary. Similarly, using standard mix cascade techniques [7], we assume the adversary cannot distinguish the correspondences of input and output messages through an anonymity system. We also assume that the other $u-1$ users in the system are honest, in that they will not provide the adversary any of the secrets they generate or use. However, the adversary also knows the distribution of their queries—a necessary assumption to model attacks based on side or background information.

Security Definitions. We define (ϵ,δ)-privacy as a flexible privacy metric to fully capture weaker as well as strong privacy-friendly search protocols. The definition relies on the following game between the adversary and honest users: an adversary provides a target user $\mathcal{U}_t$ with two queries Q_i and Q_j, and to each other user $\mathcal{U}_k$, a query Q_{n_k} from the non-target query set $\mathcal{Q}_n$, $|\mathcal{Q}_n| \leq u-1$. The target $\mathcal{U}_t$ selects one of the two queries, and uses the PIR system in order to execute it, and the others execute the corresponding Q_{n_k}, leading to the adversary observing a trace of events $\mathcal{O}$. This trace includes all information from corrupt servers and all metadata of encrypted communications from the network. We then define privacy as follows:

Privacy Definition 1. *A protocol provides* (ϵ, δ)-private PIR *if there are non-negative constants ϵ and δ, such that for any possible adversary-provided queries Q_i, Q_j, and $\langle Q_{n_k} \rangle$, and for all possible adversarial observations $\mathcal{O}$ in the observation space Ω we have that*

$$\Pr(\mathcal{O}|Q_i) \leq e^{\epsilon} \cdot \Pr(\mathcal{O}|Q_j) + \delta.$$

Note that for notational convenience, we write $\Pr(\mathcal{O}|Q_i)$ (and similarly for Q_j), showing the conditional to be on the target user's choice, even though technically the condition is over the vector $\langle Q_{n_k} \rangle$ of non-target queries as well.

In the important special case where $\delta = 0$ we call the stronger property ϵ-privacy, and can also define the likelihood ratio $\mathcal{L}$:

Privacy Definition 2. *The likelihood ratio of a particular observation $\mathcal{O}$ in an* ϵ-private PIR *scheme is defined by: $\Pr(\mathcal{O}|Q_i)/\Pr(\mathcal{O}|Q_j) \leq e^{\epsilon}$, and the likelihood ratio of the scheme itself is the maximal such value:*

$$\mathcal{L} = \max_{Q_i, Q_j, \langle Q_{n_k} \rangle, \mathcal{O}} \frac{\Pr(\mathcal{O}|Q_i)}{\Pr(\mathcal{O}|Q_j)} \leq e^{\epsilon}.$$

These definitions combine aspects of game-based cryptographic definitions and also differential privacy. We first note how the target user $\mathcal{U}_t$ may chose either Q_i or Q_j with arbitrary a-priori probability, rather than at random. The prior preference between those does not affect $\Pr(\mathcal{O}|Q_i)$ or $\Pr(\mathcal{O}|Q_j)$ that relate to the quantity to be bounded, making this definition independent of the adversary's prior knowledge of the target user's query.

Similarly the defined security game assumes that the adversary knows precisely the queries of all users except the target ($\mathcal{U} \setminus \mathcal{U}_t$), thus capturing any susceptibility to side information they would have about the queries of other users. We note that while users are provided with adversarial queries, the adversary does not learn either any user secrets created as part of the PIR protocols or the user requests sent to honest database servers.

Generality and necessity of definition. In the preferable case $\delta = 0$, the likelihood ratio of any observation is bounded, and we can therefore cap privacy leakage in all cases. A non-zero δ denotes cases where the leakage may be unbounded: events catastrophic to privacy may occur with probability at most δ. In those cases, requiring δ to be a negligible function yields a traditional computational cryptographic scheme. Note, however, that while negligible δ is sufficient to yield CPIR, not all CPIR schemes (unlike traditional IT-PIR) will have a finite ϵ, as computational indistinguishability does not imply statistical closeness.

In the case $\epsilon = 0$, we recover the cryptographic definition of indistinguishability. The traditional unconditional security provided by IT-PIR is equivalent to a mechanism with $\epsilon = 0$. For $\epsilon > 0$ information about the query selected leaks at a non-negligible rate, and users should rate-limit recurring or correlated queries as for other differentially private mechanisms [14].

Thus we lose no generality by using this definition: it can capture information-theoretic PIR systems, computational PIR systems, as well as systems that leak more information. In the rest of the paper we will define such leaky systems, making this relaxed definition necessary; we will also show that the composition of an ϵ-private PIR mechanism with an anonymity system can lead to systems that provide arbitrarily good privacy.

As for the original differential privacy definition, the ϵ-private PIR definition (with $\delta = 0$) ensures that there is no observation providing the adversary certainty to include or exclude any queries from the a-priori query set. When a PIR system does not provide such a guarantee there exist observations that allow the adversary to exclude candidate queries with certainty, leading to poor composition under repeated queries, as studied in the next section. Deciding what maximum value for epsilon is desirable depends on the sensitivity of the query and overall sought level of privacy but also usability. For instance, an epsilon lower than 10^{-2} can be considered acceptable as it implies that any particular query is only 1% more likely to have been sent than another. Furthermore, the composition of non ϵ-private PIR schemes with an anonymity channel is not guaranteed to approach perfect privacy as it may leak a lot of information to an adversary with side information about the target, or knowledge about the queries performed by other users.

3 Non ϵ-Private Systems

In this section we analyze two approaches to achieving query privacy that we show are not ϵ-private. We also examine their properties for extreme and impractical security parameters as well as when they are composed with an anonymity system.

We note that the literature does not refer to those as "Private Information Retrieval" or PIR, reserving this term for information theoretically and computationally secure schemes. Yet these ad-hoc systems fulfill the same privacy and functional role as PIR: they are used to lookup a record privately out of a public database, at a lower cost compared with IT-PIR or CPIR. Thus we examine them, analyze their properties, and use some of their ideas as ingredients to build more robust low-cost systems.

3.1 Naive Dummy Requests

A number of works attempt to hide a true user query to a single untrusted database, by hiding it among a number p of artificially generated user queries ('dummies') to achieve some privacy; for example OB-PWS [5] in the context of web search, and Hong and Landay [19] and Kido et al. [21] in the context of private location queries. Zhao et al. propose a dummy-based privacy mechanism for DNS lookups [33], but Hermann et al. find its security lacking [18]. It is interesting to note that both location and DNS applications involve large databases making traditional PIR prohibitively expensive. We show that this mechanism is not ϵ-private, leading occasionally to spectacular information leaks as reported.

Algorithm 3.1: Naive Dummy Requests (User)

Input: Query $Q \quad (0 \leq Q < n)$;
Security parameter $p \quad (p > 1)$;

$Req \leftarrow \{Q\}$;
while $|Req| < p$ **do**
  $Q' \leftarrow$ **random**(n);
  $Req \leftarrow Req \cup Q'$;
forall the $r \in Req$ **do**
  $(index_r, rec_r) \leftarrow$ **sendreceive**$(\mathcal{DS}, r)$;
return rec_Q;

The function **random**(n) samples uniformly an integer from 0 to $n-1$, and **sendreceive**(D, m) sends a message m to D, and returns any response from D.

Vulnerability Theorem 1. *The Naive Dummy Requests mechanism for security parameter $p < n$ is not ϵ-private.*

Proof. The adversary controlling the database observes which records are queried. Without loss of generality, in case the user queries for Q_i, with some probability $\mathcal{A}$ does not see the query requesting record $j : Q_j$. We denote by $\Pr(\mathcal{O}|Q_i)$ the probability an adversary $\mathcal{A}$ observes a trace of events $\mathcal{O}$ knowing the query Q_i was sent. Thereby, as the adversary has not seen the query Q_j in the current observation $\mathcal{O}$, the adversary knows the record j was not sought by the user, hence $\Pr(\mathcal{O}|Q_j) = 0$. Consequently, there is no ϵ such that $\Pr(\mathcal{O}|Q_i)/\Pr(\mathcal{O}|Q_j) \leq e^{\epsilon}$. As the ϵ-privacy bound must apply for any observation $\mathcal{O}$, and requests Q_i and Q_j, this counter example shows that the use of dummies alone does not guarantee ϵ-privacy. □

Practically, this means that if $p < n$, the adversary observing the database system DS will be able to learn, with perfect certainty, that records that have not been requested are not the sought record Q. Thus, this mechanism is not ϵ-private, until $p = n$ at which point it becomes perfectly private ($\epsilon = 0$) and corresponds to the naive download of the full database.

This weakness has practical implications: in the case of a location privacy mechanism an adversary learns which locations a user is not in with certainty, and in the context of DNS lookups, which domains are not being requested. If using this naive scheme in the context of DP5 [8], a system using PIR to protect users' social networks, an adversary would learn with certainty which social links are not present at each query.

3.2 Naive Anonymous Requests

Sending a query through an anonymity system has been proposed to maintain privacy against an untrusted database: the seminal Tor system [13] supports private queries to websites, but also performs anonymous requests as a way to resolve *dot-onion* addresses to rendezvous points. Privé [15] uses an anonymity system to query location-based services, and another proposal to perform private search engine queries [28]. However, this technique alone does not provide ϵ-private PIR.

Algorithm 3.2: Naive Anon. Request (User)

Input: Query $Q \quad (0 \leq Q < n)$;
1 $(index_Q, rec_Q) \leftarrow$ **anonsendreceive**(DS, Q);
2 **return** rec_Q;

In this mechanism users simply send requests for the records they seek to the database service through a bi-directional anonymity channel, allowing for anonymous replies (the **anonsendreceive** function). Upon receiving an anonymous request for a record, the database server simply sends the record back to the user though the anonymous channel. The hope is that since multiple queries are mixed together, the exact query of the target user is obscured. However, there is significant leakage and the mechanism in not ϵ-private.

Vulnerability Theorem 2. *The Naive Anonymous Requests mechanism is not ϵ-private, for any number of users u using the system.*

Proof. Following our game-based definition for ϵ-privacy non-target clients' queries are provided by the adversary and are all in $\mathcal{Q}_n$. As the adversary can select a non-target query set $\mathcal{Q}_n$ that does not contain Q_i or Q_j, the adversary will observe one of Q_i or Q_j only, and all other requests will be in $\mathcal{Q}_n$. For the record $Q_x, x \in \{i, j\}$ that is not queried, the probability $\Pr(\mathcal{O}|Q_x)$ equals zero, and the likelihood ratio $\mathcal{L}$ goes to infinity. Thus there exists no ϵ that may bound this likelihood. □

The proof relies heavily on the fact that the adversary provides all non-target users with known queries from the non-target query set $\mathcal{Q}_n$ and is therefore able to filter those out at the corrupt database server, and uncover trivially the target user's query. This is an extreme model; however, it also covers realistic attacks. For example, if the adversary knows that most other users are not going to access either Q_i or Q_j, but suspects that the target user might, a single observation can confirm this suspicion. This could be the case, for example, when users attempt to look up unpopular, or even personal records that only concern, and are accessed by, the target. The fact that the security parameter of the system, namely the number of users, does not affect security is particularly damning.

3.3 Composing Naive Mechanisms

Interestingly, the composition of the two naive mechanisms, namely when multiple users perform Naive Dummy Requests through an anonymous channel, for any $p > 1$, the mechanism becomes (ϵ, δ)-private. This simply involves replacing the **sendreceive** method with an anonymous channel **anonsendreceive** in the Naive Dummy Requests algorithm. As the number of users u increases the probability δ any record is requested zero or u times exactly becomes negligible and then there exists an ϵ that satisfies the definition.

More specifically, in our indistinguishability game scenario, the probability the adversary observes exactly u queries Q_i is bounded above by $\delta_u \leq \left(\frac{p-1}{n-1}\right)^{u-1}$ while the probability they receive no Q_i queries is bounded above by $\delta_0 \leq \left(\frac{n-p}{n-1}\right)^{u-1}$. (The proof can be found in Appendix A.1.) This requires a large number of users u or volume of dummies p to provide meaningful privacy against the single corrupt server. For this reason we instead explore multi-server mechanisms in the next sections.

4 Four ϵ-Private PIR Systems

4.1 Direct Requests

The first ϵ-private PIR mechanism uses dummy queries on *multiple* PIR databases, of which d_a are adversarial and $(d - d_a)$ are honest. The user generates a query for the sought record, along with $p - 1$ random (distinct) other ones. The requests are partitioned into sets of equal size and sent to the PIR databases directly. Each database then responds with the list of records requested, encrypted as are all communications.

The database servers simply respond to requests by returning the index and the records sought over the encrypted

Algorithm 4.1: Direct Requests (User)

Input:
Q: $(0 \leq Q < n)$;
p: $(p > 1) \wedge p \equiv 0 \mod d$;

$Req \leftarrow \{Q\}$;
while $|Req| < p$ **do**
  $Q' \leftarrow$ **random**(n);
  if $Q' \notin Req$ **then**
    $Req \leftarrow Req \cup Q'$;
for $1 \leq i \leq d$ **do**
  for $1 \leq j \leq p/d$ **do**
    $r \leftarrow$ **pop**(Req)
    $(index_r, rec_r) \leftarrow$ **sendreceive**(DB_i, r);
return rec_Q;

channel. **pop**(Req) returns a random item from the set Req (and also removes it from Req). rec_Q is the sought record of index Q.

Security Theorem 1. *The* direct requests mechanism *is an ϵ-private PIR mechanism with*

$$\epsilon = \ln\left(\frac{1}{d-d_a} \cdot \left(d \cdot \frac{n-1}{p-1} - d_a\right)\right),$$

where d is the number of databases, of which d_a are adversarial, n is the total number of records, and p is the total number of queries sent by the user.

Proof. See Appendix A.2. □

Costs. The client needs to communicate p/d record indices to each of d servers. How to best do this depends on the size of p/d. If p/d is small, the best way is simply to send the p/d indices (each of size $\lceil \lg n \rceil$ bits) to each of d servers, for a total cost of $p \cdot \lceil \lg n \rceil$ bits. However, if p/d is large, it is more efficient to simply send an n-bit bitmask to each server, for a total cost of $d \cdot n$ bits. Therefore the total client communication is $C_c = \min(d \cdot n, p \cdot \lceil \lg n \rceil)$ bits. The server communication cost is just $C_s = p$, as p records are requested and sent back to the user. As the databases do not XOR any records but just accesses and sends them, the computation cost is $C_p = p \cdot c_{acc}$.

Practical values. Fig. 1 illustrates Direct Request curves representing ϵ as a function of p for different adversaries in the reference scenario of Certificate Transparency. As millions of certificates have already been recorded in databases and hundreds of databases are supposed to be running all over the world, we have set $n = 10^6$ and $d = 10^2$. The security parameter ϵ starts above 10 and slowly diminishes until nearly all of the records have been requested where the curves follow

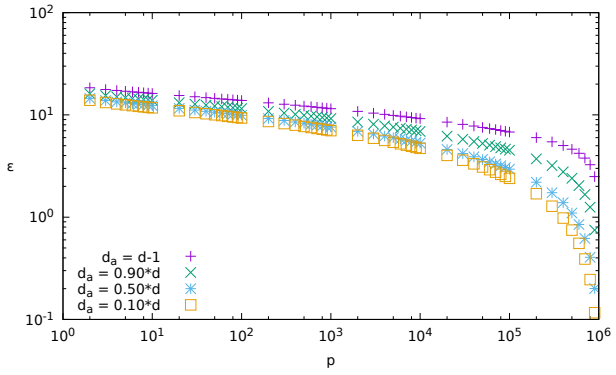


Fig. 1. Direct requests: ϵ versus p for $d = 100$ and $n = 10^6$

the vertical asymptote $p = n$. If weaker adversaries decrease ϵ for any p, the difference becomes really noticeable only after requesting a tenth of the records. Further, in order to achieve even a mediocre security of $\epsilon < 1$, for any d_a, more than $\frac{9}{10}$ of the records have to be requested. In the worst-case scenario where only one database is not colluding, we find the security parameter ϵ is approximately equal to 11.5. However if only half of the databases are corrupted, i.e. $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 7.6$. To summarize for $n = 10^6$, $d = 10^2$ and $p = 10 \cdot d$, if $d_a = d - 1$ we have $\epsilon \approx 11.5$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 7.6$. For any d_a, to obtain $\epsilon < 1$, $p > \frac{9}{10} \cdot n$.

In the case of a small database system consisting of a few to tens of databases, each storing thousands of records, we set $n = 10^3$ and $d = 10$. When the adversary controls all databases but one, if the user only sends one request per database we have that $\epsilon \approx 7$ while when half of the databases are corrupted, $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 5.4$. To summarize for $n = 10^3$, $d = 10$ and $p = d$, if $d_a = d - 1$ we have $\epsilon \approx 7$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 5.4$.

The above examples illustrate that for large databases, as the one considered in the motivating Certificate Transparency example, an adversary controlling about half the databases can extract a lot of information. Furthermore, information leakage does not diminish significantly based on the security parameter p, or for smaller databases. Thus we conclude the Direct Requests mechanism alone provides very weak privacy; however, we will show how its composition with an anonymity system can improve its performance.

4.2 Anonymous direct requests

Bundled anonymous request

We compose the *direct requests* mechanism from the previous subsection with an anonymous channel. Each user, including the target user $\mathcal{U}_t$, sends a bundle of requests defined by the *direct requests* PIR mechanism to databases through an anonymity system $\mathcal{AS}$.

The requests are *bundled*, in that all requests from a specific user are linkable with each other, allowing this mechanism to be implemented by sending a single anonymous message through the $\mathcal{AS}$ per user. The $\mathcal{AS}$'s exit node receiving the bundle forwards the different sets of queries (as usual, encrypted by the user to each respective database) to the relevant database and anonymously returns the requested records from each database.

The increased privacy of this scheme derives from the ability of the target user $\mathcal{U}_t$ to hide the use of the PIR system amongst $u-1$ other users. This strengthens the direct requests mechanism hiding $\mathcal{U}_t$'s query amongst $p-1$ random requests throughout d servers. The adversary's task becomes harder as any bundle, out of u, could be the target's, and any query, out of p, the correct one.

When seeing one of the non-target queries Q_{n_k}, the adversary can however link the corresponding bundle to a non-target user $\mathcal{U}_k$ with overwhelming probability, discard it, and thus reduce the his analysis to fewer users. Not to discard the target's bundle, the adversary should minimize the probability the target user chooses an element of $\mathcal{Q}_n$, which equals $(p-1)\cdot\frac{|\mathcal{Q}_n|}{n}$. To do so, the adversary thus should send the same non-target query Q_{n_k} to all non-target users, creating a non-target query set of size 1; that is, simply choosing $\mathcal{Q}_n = \{Q_0\}$ would result in the best observation for the adversary.

The database servers simply respond to bundles by returning the index and the records sought over the encrypted channel, the anonymity system forwarding the answer to the corresponding users.

Security Theorem 2. *The* bundled anonymous requests mechanism *is ϵ-private with*

$$\epsilon = \ln\left(\left(\frac{d}{d-d_a}\cdot\frac{n-1}{p-1} - \frac{d_a}{d-d_a}\right)^2 + u - 1\right) - \ln u.$$

Proof. By applying our Composition Lemma (see below), and the security parameter of the direct requests mechanism. □

Costs. As the only differences with the Direct Request case is the Anonymity system and the bundling of the messages, we find the same values for the communication costs $C_c = \min(d\cdot n, p\cdot\lceil\lg n\rceil)$ and $C_s = p$, and the computation cost $C_p = p\cdot c_{acc}$.

Algorithm 4.2: Bundled Anonymous Requests (User)

Input:
Q: $(0 \le Q < n)$;
p: $(p > 1) \wedge p \equiv 0 \mod d$;

$Req \leftarrow \{Q\}$;
while $|Req| < p$ **do**
  $Q' \leftarrow$ **random**(n);
  if $Q' \notin Req$ **then**
    $Req \leftarrow Req \cup Q'$;
$Bundle \leftarrow \{\}$
for $1 \le i \le d$ **do**
  for $1 \le j \le p/d$ **do**
    $Bundle_i \leftarrow$ **pop**(Req)
  $Bundle \leftarrow (DB_i, Bundle_i)$
$(index_r, rec_r) \leftarrow$ **anonsendreceive**$(DS, Bundle)$;
return rec_Q;

Separated anonymous request

We may also compose the *direct requests* mechanism (Sect. 4.1) with an anonymous channel in a different manner. Each user, including the target user $\mathcal{U}_t$, sends distinct requests defined by the *direct requests* PIR mechanism to databases through an anonymity system $\mathcal{AS}$, whose queries are *unlinkable* at the mix output.

The requests are *separated*, in that all requests from a specific user are unlinkable with each other, allowing this mechanism to be implemented by sending separate anonymous messages through the $\mathcal{AS}$ to different databases. The $\mathcal{AS}$'s exit node receiving the message forwards it to the relevant database and anonymously returns the requested record.

The increased privacy of this scheme derives from the ability of the target user $\mathcal{U}_t$ to hide the real query of the PIR system amongst $u\cdot(p-1)$ other random queries.

Algorithm 4.3: Separated Anonymous Requests (User)

Input:
Q: $(0 \le Q < n)$;
p: $(1 < p) \wedge p \equiv 0 \mod d$;

$Req \leftarrow \{Q\}$;
while $|Req| < p$ **do**
  $Q' \leftarrow$ **random**$(0, n)$;
  if $Q' \notin Req$ **then**
    $Req \leftarrow Req \cup Q'$;
forall the $i \in p$ **do**
  $r \leftarrow$ **pop**(Req);
  $(index_r, rec_r) \leftarrow$ **anonsendreceive**(DS_i, r);
return rec_Q;

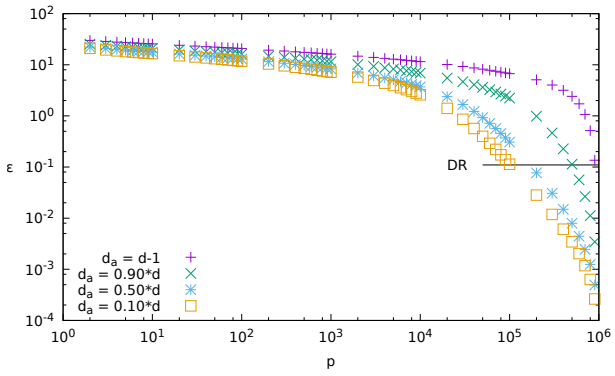


Fig. 2. AS-Bundle: ϵ versus p for $d = 100$, $n = 10^6$, and $u = 10^3$

Since the Bundled Anonymous Requests mechanism leaks strictly more information than Separated Anonymous Request, the ϵ_{bundle} is also an upper bound of the Separated Anonymous Request.

Costs. As this method is similar with the Bundled case, we have for costs $C_c = \min(d \cdot n, p \cdot \lceil \lg n \rceil)$, $C_s = p$ and $C_p = p \cdot c_{acc}$. However, the load on the anonymity system increases as there are $u \cdot p$ anonymous messages transmitted.

Practical values. Fig. 2 shows Direct Request composed with anonymity system curves representing ϵ as a function of p for different adversaries in the reference scenario of Certificate Transparency. As before, we set $n = 10^6$ and $d = 10^2$ and assumed $u = 10^3$. The security parameter ϵ starts above 10 and slowly diminishes until a tenth to most of the records have been recorded depending on d_a where the curves follow the vertical asymptote $p = n$. The anonymity system gain in privacy can be seen under the line indicating where the privacy of the Direct Request protocol, without an anonymity system, stops for the same amount of points. If the anonymity system gains appear negative at the beginning of the curves, this is due to the lack of tightness of the bound in the Composition Lemma. If weaker adversaries decrease ϵ for any p, the difference becomes really noticeable only after requesting a hundredth of the records. Further, in order to achieve even a mediocre security of $\epsilon < 1$, for any d_a, at most half of the records have to be requested compared to 90% without an anonymity system. In the worst-case scenario where only one database is not colluding, we find the security parameter ϵ is approximately equal to 16. However if only half of the databases are corrupted, i.e. $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 8$. To summarize for $n = 10^6$, $d = 10^2$, $u = 10^3$ and $p = 10 \cdot d$, if $d_a = d - 1$ we have $\epsilon \approx 16$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 8$.

In the case of a small database system managing a few to tens of databases, each storing thousands of records, we again set $n = 10^3$ and $d = 10$. When the adversary controls all databases but one, each sending only one request per database, we have that $\epsilon \approx 7$ while when half of the databases are corrupted, $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 4$. To summarize for $n = 10^3$, $d = 10$, $u = 10^3$ and $p = d$, if $d_a = d - 1$ we have $\epsilon \approx 7$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 4$.

We conclude that direct requests through an anonymity system is a stronger mechanism that direct requests alone. However, for very large databases, such as the one expected in Certificate Transparency, the quality of protection is still low. It becomes better only as the total volume of requests from all users is in the order of magnitude of the number of records in the database. This requires either a large number of users, or a large number of dummy requests per user. However, even the weaker protection afforded by anonymous direct requests may be sufficient to protect privacy in applications where records only need to be accessed infrequently.

4.3 Sparse-PIR

We next adapt Chor's simplest IT-PIR scheme [10] to reduce the number of database records accessed to answer each query. As a reminder: in Chor's scheme the user builds a set of random binary vectors of length n (the number of records in the database), one for each server; we call these vectors the "request vectors". These are constructed so that their element-wise XOR yields a zero for all non-queried records, and a one for the record sought (we call this the "query vector"). Each server simply XORs all records corresponding to a 1 in its request vector, and returns this value to the user. The XOR of all responses corresponds to the sought record.

Sparse-PIR aims to reduce the computational load on the database servers $\mathcal{DB}_i$. To this end the binary request vectors are not sampled uniformly but are sparse, requiring the database servers to access and XOR fewer records to answer each query. Specifically, in Sparse-PIR each request is derived by independently selecting each binary element using a Bernoulli distribution with parameter $\theta \leq 1/2$. Furthermore, the constraint that the XOR of these sparse vectors yields the query vector is maintained. The intuition is that we will build a $d \times n$ query matrix M *column wise*: each column (of length d) corresponds to one record in the database, and will be selected by performing d independent Bernoulli trials with parameter θ, re-sampling if necessary to ensure the sum of the entries in the column (the Hamming weight) is even for non-queried records, or odd for the single queried record.

Equivalently, we may first select a Hamming weight for each column with the appropriate probability depending on d,

θ, and whether the column represents the queried record or not, and then select a uniformly random vector of length d with that Hamming weight. Each row of the query matrix will then have expected Hamming weight $\theta \cdot n$, and the rows of the matrix (the request vectors) will XOR to the desired query vector, namely all 0 except a single 1 at the desired location.

Algorithm 4.4: Sparse-PIR (User)

Input:
Q: $0 \leq Q < n$;
θ: $0 < \theta \leq \frac{1}{2}$;

$M \leftarrow [\,]$;
for $0 \leq col < n$ **do**
  if $col = Q$ **then**
    $q \leftarrow d$ Bernoulli(θ) trials with Odd sum;
  else
    $q \leftarrow d$ Bernoulli(θ) trials with Even sum;
  $M \leftarrow M$ **append column** q;
for $1 \leq i \leq d$ **do**
  $r_i \leftarrow$ **row** i **of** M;
  $resp_i \leftarrow$ **sendreceive**(DB_i, r_i);
return $\bigoplus_{1 \leq i \leq d} resp_i$;

The database logic in Sparse-PIR is identical to the logic in Chor's IT-PIR: each database server receives a binary vector, XORs all records that correspond to entries with a 1, and responds with the result. In fact the database may be agnostic to the fact it is processing a sparse PIR request, aside from the reduction in the number of entries to be XORed. For $\theta < 0.5$ the costs of processing at each database is lowered due to the relative sparsity of ones, at no additional networking or other costs.

Security Theorem 3. *The Sparse-PIR mechanism is ϵ-private with*

$$\epsilon = 4 \cdot \operatorname{arctanh}[(1-2\theta)^{(d-d_a)}],$$

where θ is the parameter of the Bernoulli distribution and $d - d_a$ represents the number of honest PIR servers.

Proof. See Appendix A.3. □

As expected when $\theta = 1/2$ the privacy provided by the Sparse-PIR mechanism is the same as for the perfect IT-PIR mechanism. This fact can be derived from the tight bound on ϵ by observing that ϵ equals zero when $\theta = 1/2$.

Security Lemma 1. *For $\theta = 1/2$, and at least one honest server, the Sparse-PIR mechanism provides perfect privacy, namely with $\epsilon = 0$.*

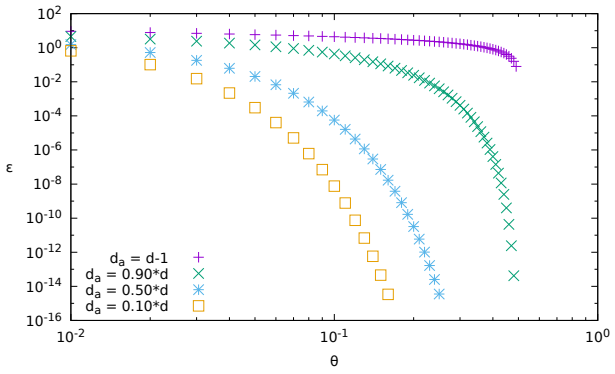


Fig. 3. Sparse-PIR: ϵ versus θ for $d = 100$

More interestingly, as the number of honest servers increases, the privacy of the Sparse-PIR increases for any θ, and in the limit becomes perfect as in standard IT-PIR:

Security Lemma 2. *For an increasing number of honest servers $(d - d_a) \to \infty$ the Sparse-PIR mechanism approaches perfect privacy, namely $\epsilon \to 0$.*

Proof. Note that for $0 < \theta < 1$, $\lim_{x\to\infty}(1-2\theta)^x = 0$. Thus for $(d - d_a) \to \infty$ we have that $\epsilon \to 0$, since $\operatorname{arctanh}(0) = 0$. □

Costs. In Sparse-PIR, the client requests the XOR of $\theta \cdot n$ records from each of the d servers. As above, how best to do this depends on the size of θ. If $\theta > \frac{1}{\lg n}$, then sending an n-bit bitmask to each of the d servers will do (just as in the standard Chor case, where effectively $\theta = \frac{1}{2}$); if θ is smaller, however, then listing the $\theta \cdot n$ indices, at a cost of $\theta \cdot n \lceil \lg n \rceil$ bits for each of the d servers is cheaper. The total client communication cost is therefore $C_c = d \cdot \min(n, \theta \cdot n \lceil \lg n \rceil)$. On each of the d servers, only $\theta \cdot n$ records are accessed and operated on per request, and a single record is sent. We thus have $C_s = d$ and $C_p = \theta \cdot d \cdot n \cdot (c_{acc} + c_{prc})$.

Practical values. Fig. 3 shows Sparse-PIR curves representing ϵ as a function of θ for different adversaries in the reference scenario of Certificate Transparency with $d = 10^2$. The security parameter ϵ starts below 10 and slowly diminishes until nearly all of the records have been accessed for $\theta = \frac{1}{2}$ where the curves follow a vertical asymptote. The difference in ϵ for different adversaries is noticeable at any point of the curves. In order to achieve even a mediocre security of $\epsilon < 1$, except for the worst case $d_a = d - 1$, accessing 10% of the records at each database is enough. In the worst-case scenario where only one database is not colluding, we find the security parameter ϵ is approximately equal to 2 for $\theta = 0.25$. However

if only half of the databases are corrupted, i.e. $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 10^{-15}$ for the same θ. To summarize for $d = 10^2$ and $\theta = 0.25$, if $d_a = d - 1$ we have $\epsilon \approx 2$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 10^{-15}$.

In the case of a small database systems managing a few to tens of databases, we set $d = 10$. When the adversary controls all databases but one, we have the $\epsilon \approx 2$ while when half of the databases are corrupted, $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 10^{-1}$. To summarize for $d = 10$ and $\theta = 0.25$, if $d_a = d - 1$ we have $\epsilon \approx 2$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 10^{-1}$.

A sparse version of the simple Chor scheme can indeed protect the user's privacy better than the direct request, as we can observe a factor of 9 between the two epsilons. Yet, in the worst-case scenario, where the adversary controls all the databases except one, the risk is still significant: the adversary infers that the user is about 7 times more likely to seek a particular record over another. Thus we consider strengthening the system through composition with an anonymous channel.

4.4 Anonymous Sparse-PIR

We consider the composition of the Sparse-PIR mechanism with an anonymity system. In this setting, a number of users u select their queries to the database servers, and perform them anonymously through an anonymity system. We consider that all requests from the same user are linkable to each other at the input and output of the anonymity system. As per our standard setting, the adversary provides a target user $\mathcal{U}_t$ with queries Q_i and Q_j, one of which the user chose, and all other $u - 1$ users with $Q_{n_k} \in \mathcal{Q}_n$. They all use an *arbitrary* ϵ-private PIR mechanism through an anonymity channel to perform their respective queries.

We will show that this mechanism is ϵ-private, through first proving a general composition lemma. This could be of independent interest to designers of private query systems based on anonymous channels.

Composition Lemma. *The composition of an arbitrary ϵ_1-private PIR mechanism with a perfect anonymity system used by u users, for sufficiently large u, yields an ϵ_2-private PIR mechanism with:*

$$\epsilon_2 = \ln(e^{2\epsilon_1} + u - 1) - \ln u.$$

Proof. See Appendix A.4. Note this is not a worst-case analysis, but an average-case analysis over the honest users' randomness, which can not be influenced by the adversary. Namely there is a negligible probability in u, the number of users in the anonymity system, this does not hold. A fuller (ϵ, δ)-privacy definition could capture the worst-case behaviour. □

Algorithm 4.5: Anonymous Sparse-PIR (User)

Input:
Q: $0 \leq Q < n$;
θ: $0 < \theta \leq \frac{1}{2}$;

$M \leftarrow [\,]$;
for $0 \leq col < n$ **do**
  if $col = Q$ **then**
    $q \leftarrow d$ Bernoulli(θ) trials with Odd sum;
  else
    $q \leftarrow d$ Bernoulli(θ) trials with Even sum;
  $M \leftarrow M$ **append column** q;
for $1 \leq i \leq d$ **do**
  $r_i \leftarrow$ **row** i **of** M;
  $resp_i \leftarrow$ **anonsendreceive**(DB_i, r_i);
return $\bigoplus_{1 \leq i \leq d} resp_i$;

It is easy to show that as $u \to \infty$, the parameter $\epsilon_2 \to 0$, leading to a perfect IT-PIR mechanism, independently of the value of ϵ_1 (so long as it is finite). Conversely, when $u = 1$, we have $\epsilon_2 = 2\epsilon_1$ (the loss of a factor of 2 is due to the lack of tightness of the bound). Using this lemma, we can prove our main theorem.

Security Theorem 4. *The composition of the Sparse-PIR scheme with parameters θ, d, and d_a with an anonymity system with u users is also ϵ-private with security parameter*

$$\epsilon = \ln\left(\left(\frac{1 + (1-2\theta)^{(d-d_a)}}{1 - (1-2\theta)^{(d-d_a)}}\right)^4 + u - 1\right) - \ln u.$$

Proof. This is a direct consequence of the Composition Lemma, the security parameter of Sparse-PIR, and the definition of $\operatorname{arctanh}$. □

Cost The use of an anonymity system does not change any of the communication or computation costs. The communication costs remain $C_c = d \cdot \min(n, \theta \cdot n \lceil \lg n \rceil)$ and $C_s = d$, and the computation cost remains $C_p = \theta \cdot d \cdot n \cdot (c_{acc} + c_{prc})$.

Practical values. Fig. 4 shows Sparse-PIR composed with anonymity system curves representing ϵ as a function of θ for different adversaries in the reference scenario of Certificate Transparency, with $d = 10^2$ and $u = 10^3$. The security parameter ϵ starts below 10 and slowly diminishes until nearly all of the records have been accessed for $\theta = \frac{1}{2}$ where the curves follow a vertical asymptote. If the anonymity system gains appear negative at the beginning of the curves, this is due to the lack of tightness of the bound in the Composition Lemma. The

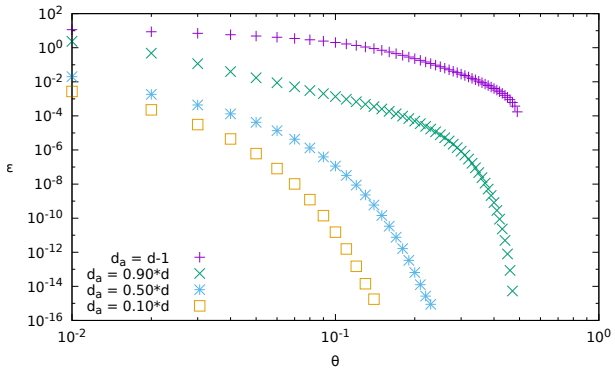


Fig. 4. AS-Sparse-PIR: ϵ versus θ for $d = 100$ and $u = 10^3$

difference in ϵ for different adversary is noticeable at any point of the curves. In order to achieve even a mediocre security of $\epsilon < 1$, except for the worst case $d_a = d - 1$, accessing more than 10% of the records at each database is enough. In the worst-case scenario where only one database is not colluding, assuming there are 1000 users, we find the security parameter ϵ is approximately equal to 10^{-1}. However, if only half of the databases are corrupted (i.e., $d_a = \frac{1}{2} \cdot d$), we have $\epsilon < 10^{-15}$. To summarize for $d = 10^2$, $u = 10^3$ and $\theta = 0.25$, if $d_a = d - 1$ we have $\epsilon \approx 10^{-1}$ while if $d_a = \frac{d}{2}$, we have $\epsilon < 10^{-15}$.

In the case of a small database system managing a few to tens of databases we set $d = 10$. When the adversary controls all databases but one, if there are 1000 users, each sending only one request per database, we have the $\epsilon \approx 10^{-1}$ while when half of the databases are corrupted, $d_a = \frac{1}{2} \cdot d$, we have $\epsilon \approx 10^{-3}$. To summarize for $d = 10$, $u = 10^3$ and $\theta = 0.25$, if $d_a = d - 1$ we have $\epsilon \approx 10^{-1}$ while if $d_a = \frac{d}{2}$, we have $\epsilon \approx 10^{-3}$.

Anonymous Sparse-PIR allows us to easily trade off θ (which governs the server-side cost of the protocol) with u (the number of simultaneous users of the database). If the number of users is high, then by composing Sparse-PIR with an anonymity system, we can reduce θ and still achieve a low ϵ.

5 Optimizing PIR

In this section, we propose an optimization for PIR systems to render them more scalable, but at a higher risk.

5.1 Subset-PIR

In order to lower both the communication and computation costs, when $d \gg 1$, one could consider doing IT-PIR on a subset of just t of the databases. We call this optimization *Subset-PIR*. This optimization applies to any IT-PIR protocol, so long as that protocol can be used with a *client-selected* number of replicated servers. Chor's protocol is an example of such a flexible IT-PIR protocol.

The communication and server side computation costs are thus multiplied by a factor of $\frac{t}{d}$ at the cost of a greater risk of all contacted databases being compromised. Consequently, even if an IT-PIR scheme were perfectly private, this optimization induces a non-zero probability of the adversary being able to breach it.

Algorithm 5.1: Subset-PIR (User)

Input:
Q: $0 \leq Q < n$;
t: $2 \leq t \leq d$;

for $1 \leq j \leq t-1$ **do**
  $P_j \leftarrow n$ Bernoulli($\frac{1}{2}$) trials;
// e_Q is the vector with all 0s except a 1 at position Q
$P_t \leftarrow \left(\oplus_{j=1}^{t-1} P_j\right) \oplus e_Q$;
$DB \leftarrow \{\}$;
while $|DB| \leq t$ **do**
  server $\leftarrow$ **random**(d);
  if *server* $\notin DB$ **then**
    $DB \leftarrow DB \cup \{\text{server}\}$;
for $1 \leq j \leq t$ **do**
  $resp_j \leftarrow$ **sendreceive**$(DB_{DB[i]}, P_j)$;
return $\bigoplus_{i \in t} resp_i$;

Security Theorem 5. *Subset-PIR is an (ϵ, δ)-private PIR optimization with*

$$\epsilon = 0 \text{ and } \delta = \prod_{i=0}^{t-1} \frac{d_a - i}{d - i}$$

where d is the number of databases, of which d_a are compromised, and t represents the number of PIR servers contacted. When $t > d_a$ the mechanism provides unconditional privacy.

Proof. The probability of contacting t databases out of which t_a are compromised, knowing that there are in total d_a compromised databases out of d is:

$$\Pr(t_a, t \mid d_a) = \frac{\binom{d_a}{t_a} \cdot \binom{d-d_a}{t-t_a}}{\binom{d}{t}}$$

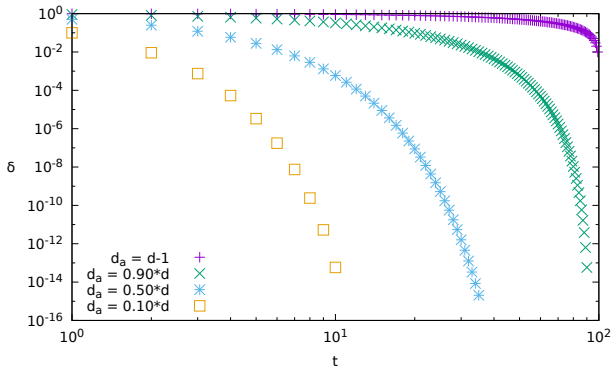


Fig. 5. Subset-PIR: δ versus t for $d = 100$

The probability of contacting only compromised databases is obtained by setting $t_a = t$, and so is $\frac{\binom{d_a}{t}}{\binom{d}{t}}$, which equals $\prod_{i=0}^{t-1} \frac{d_a - i}{d - i}$ if $t \leq d_a$, and 0 if $t > d_a$. □

Costs. For Subset-PIR, as we contact t databases, we have $C_s = t$ and using a Chor-like PIR protocol we have the computation cost $C_p = \frac{1}{2} \cdot t \cdot n \cdot (c_{acc} + c_{prc})$.

Practical values. Fig. 5 shows Subset-PIR curves representing δ as a function of the number of databases contacted t for different adversaries in the reference scenario of Certificate Transparency, with $d = 10^2$. The security parameter δ starts between 10^{-1} and 1 and slowly diminishes until a tenth to most of the databases have been contacting depending on d_a where the curves follow a vertical asymptote at $t = d$. The difference in δ for different adversaries is noticeable at any point of the curves. In order to achieve even a mediocre security of $\delta < 10^{-1}$, excluding the worst case $d_a = d - 1$, less than 20% of the databases have to be contacted. In the worst-case scenario where only one database is not colluding assuming the user contacts only a tenth of the databases, we find the security parameter δ is approximately equal to 0.9. However if only half of the databases are corrupted (i.e., $d_a = \frac{1}{2} \cdot d$), we have $\delta \approx 10^{-4}$. To summarize for $d = 10^2$ and $t = \frac{1}{10} \cdot d$, if $d_a = d - 1$ we have $\delta \approx 0.9$ while if $d_a = \frac{d}{2}$, we have $\delta \approx 10^{-4}$.

In the case of a small database system managing a few to tens of databases, each storing thousands of records, we set $d = 10$. When the adversary controls all databases but one, if the user contacts a tenth of the databases, we have that $\delta \approx 0.9$ while when half of the databases are corrupted, $d_a = \frac{1}{2} \cdot d$, we have $\delta \approx 0.5$. To summarize for $d = 10$ and $t = \frac{1}{10} \cdot d$, if $d_a = d - 1$ we have $\delta \approx 0.9$ while if $d_a = \frac{d}{2}$, we have $\delta \approx 0.5$.

Perfectly private ($\epsilon = 0$) IT-PIR designs used in conjunction with the Subset-PIR optimization become (ϵ, δ)-private with $\epsilon = 0$ and δ reasonably small, if the number of honest database servers is large. Indeed, Subset-PIR is still perfectly private, with $\epsilon = \delta = 0$, if the number of servers contacted (t) exceeds the number of adversarial servers (d_a).

Demmler et al. [12] explore a similar idea with their RAID-like design of a PIR system. In RAID-PIR, rather than the client only contacting a subset of the servers, it instead divides the database and the queries into *chunks*, and sends the query chunks corresponding to database chunk i to just t of the servers. By systematically picking which t servers to use for each chunk, however, RAID-PIR does contact all d servers, but each server only does t/d of the work it would ordinarily do, and indeed, each server need only store a t/d fraction of the database, if all clients are required to use the same value of t. Using only this feature of RAID-PIR yields a scheme with the same communication and computation costs as Subset-PIR. RAID-PIR goes further, however, and proposes to generate most of the random query values using a PRNG rather than a truly random string. This greatly reduces the client-to-server communication cost of RAID-PIR, at the cost of changing RAID-PIR from an IT-PIR protocol to a CPIR protocol [16, footnote 1].

6 Comparative Evaluation

In Table 1, we summarize for each scheme presented the security parameters ϵ and δ, the communication costs C_s, the number of blocks sent back to the user, and the computational cost C_p which depends on the access cost c_{acc} and the processing cost c_{prc}; i.e., the cost associated to the number of records XORed.

When the protocols are not fully private (i.e., $\epsilon \neq 0$), we observe a reduction in the server computation costs. The Sparse-PIR scheme diminishes the computation cost by a factor of $2 \cdot \theta$ compared to Chor PIR [10], while the Direct Request schemes induce no record processing. As the use of an anonymity system raises the privacy level, the security parameter can be lowered to reach the same privacy level of the schemes at the cost of network delays. The Sparse-PIR methods do not influence the communication cost, but the Direct Request schemes drastically increase it as the number of requests p is a multiple of d.

The Subset-PIR optimization schemes helps scalability by reducing all costs by a factor of $\frac{t}{d}$, but turns ϵ-private protocols into (ϵ, δ)-private ones.

The two main approaches for decreasing the computation are contacting fewer databases and accessing (or processing) fewer records per server. It can be noted, for example, that in order for Sparse-PIR to achieve a similar level of computation

	ϵ	δ	C_c	C_s	C_p
Chor PIR [10]	0	0	$d \cdot n$	d	$\frac{1}{2} \cdot d \cdot n \cdot (c_{acc} + c_{prc})$
Direct Requests	$\ln\left(\frac{1}{d-d_a} \cdot \left(d \cdot \frac{n-1}{p-1} - d_a\right)\right)$	0	$\min\left(d \cdot n,\, p \lceil \lg n \rceil\right)$	p	$p \cdot c_{acc}$
Sparse-PIR	$4 \cdot \operatorname{arctanh}[(1-2\theta)^{(d-d_a)}]$	0	$d \cdot \min\left(n,\, \theta \cdot n \lceil \lg n \rceil\right)$	d	$\theta \cdot d \cdot n \cdot (c_{acc} + c_{prc})$
$\mathcal{AS}$-Request	$\ln\left(\frac{1}{u}\left(\frac{d}{d-d_a} \cdot \frac{n-1}{p-1} - \frac{d_a}{d-d_a}\right)^2 + \frac{u-1}{u}\right)$	0	$\min\left(d \cdot n,\, p \lceil \lg n \rceil\right)$	p	$p \cdot c_{acc}$
$\mathcal{AS}$-Sparse-PIR	$\ln\left(\frac{1}{u}\left(\frac{1+(1-2\theta)^{(d-d_a)}}{1-(1-2\theta)^{(d-d_a)}}\right)^4 + \frac{u-1}{u}\right)$	0	$d \cdot \min\left(n,\, \theta \cdot n \lceil \lg n \rceil\right)$	d	$\theta \cdot d \cdot n \cdot (c_{acc} + c_{prc})$
Subset-PIR	0	$\prod_{i=0}^{t} \frac{d_a - i}{d - i}$	$t \cdot n$	t	$\frac{1}{2} \cdot t \cdot n \cdot (c_{acc} + c_{prc})$

Table 1. Security and Cost Summary of the Schemes. The client communication C_c is measured in bits, while the server communication C_s is measured in units of b-bit records.

to Subset-PIR with a given t, the parameter θ must be particularly low, $\theta = \frac{t}{4 \cdot d}$. The first approach would be relevant in the case of a quasi-trusted database system while the second in the case of a large untrusted one.

In Figure 6, we compare the computation cost C_p, the number of records accessed, and the communication cost C_s, the number of records sent, of the Direct Request and Sparse-PIR schemes, and their compositions with an anonymity system, for a system comparable to Certificate Transparency when the adversary controls half of the databases. If the costs of the designs with an anonymity system first appear greater than in the simple case, this can be explained by the lack of tightness of the bound in the Composition Lemma. The gains of the anonymity system can be seen by the values ϵ takes under the lines "DR" and "SP" which represent the last security value respectively for the Direct Request and Sparse-PIR designs without an anonymity system.

In Figures 6a and 6c, we show the privacy parameter ϵ as a function of the whole database system computation cost C_p and compare it between the two PIR designs and their composition with an anonymity system. For the Direct Request cases, C_p represents the total number of records accessed p while for Sparse-PIR ones this is the sum of the records accessed by each database $\theta \cdot d \cdot n$. This difference is worth mentioning as by definition a record can be accessed and sent only once in the Direct Request cases, while in the Sparse-PIR ones, a record can be accessed and processed at different servers. Thus, the privacy level will converge to 0 for $p = d$ with the Direct Request protocols but for $\theta = \frac{1}{2}$, or $p = \frac{1}{2} \cdot d \cdot n$ in the graphs, with the Sparse-PIR protocols. While both figures show ϵ decreasing with C_p, the Direct Request protocols perform better for a given C_p than the Sparse-PIR ones which however appear more flexible as the security parameter ϵ can be selected in a wider interval.

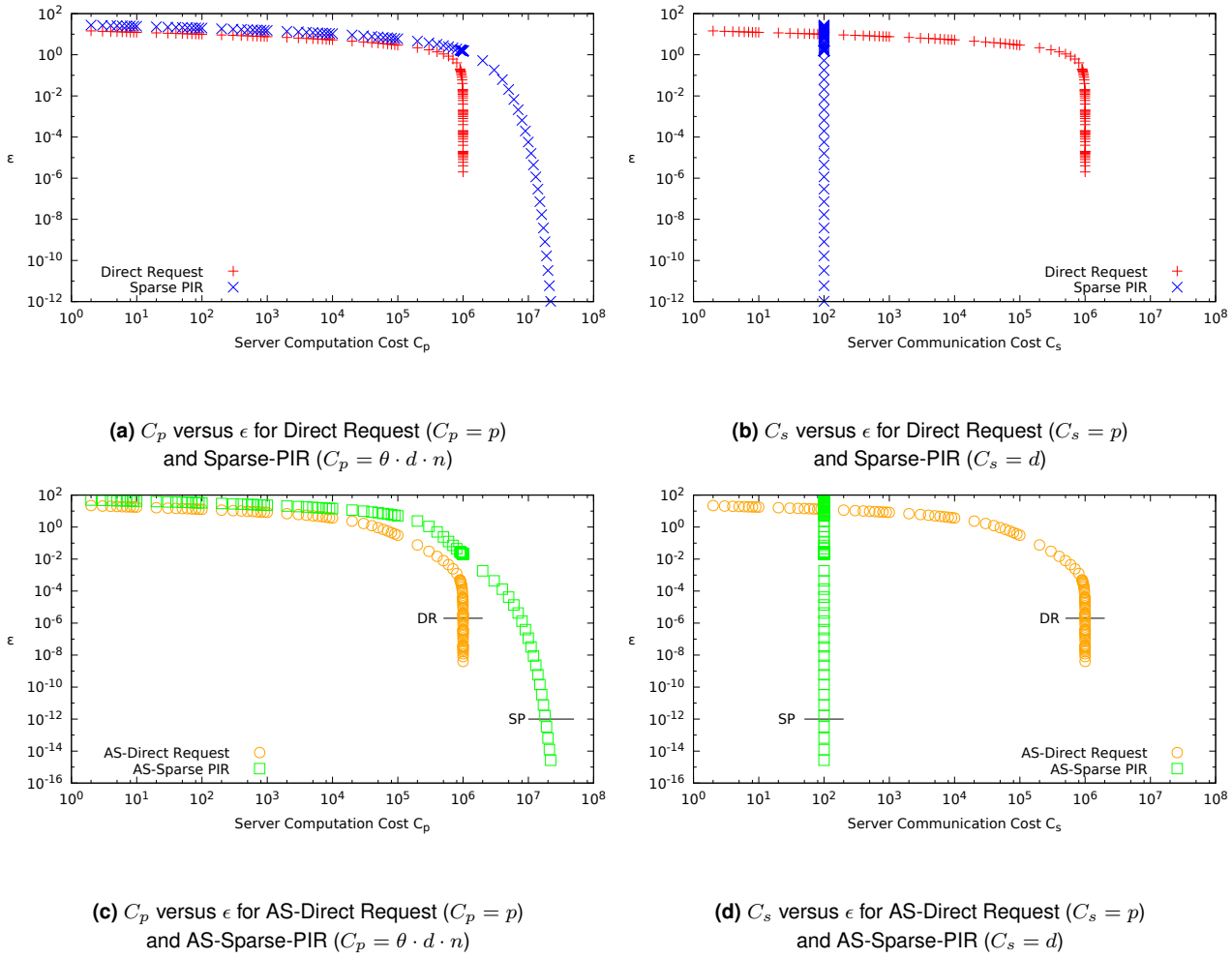
In Figures 6b and 6d, we show ϵ as a function of the number of records sent back by the whole database system to the user and compare it between the PIR designs and their compositions with an anonymity system. While the privacy level does not depend on C_s for the Sparse-PIR protocols, as the number of requests sent and record received is a constant, C_s has to greatly increase to reach an adequate ϵ in the Direct Request cases.

While the Direct Requests protocols present lower computational costs than the Sparse-PIR ones, they vastly increase the communication costs. This is not a surprise as PIR was conceived in order to limit the communication cost of private search in public databases. Choosing which method to use thus depends on the database system characteristics, not only the number of database servers and the level of trust the user has, but also the hardware. One method can be used to counter the system bottleneck, Sparse-PIR would suit servers with fast processors while Direct Request would adapt better with high-speed networks. As both processing and networking capabilities are continually increasing, the question of whether Direct Request schemes have a future is still open.

6.1 Discussion

6.1.1 Sybil Attacks

In a $n-1$, or Sybil, attack, an adversary acquires a disproportionately large influence on a system by creating a large number of identities. In this work, such an attack would translate when using the anonymity system to the adversary controlling or being most to all of the non-target clients. As a result, the number of honest users is drastically diminished, and so the adversary can guess with higher probability which queries are the target's. In the worst case when there are no honest non-



(a) C_p versus ϵ for Direct Request ($C_p = p$) and Sparse-PIR ($C_p = \theta \cdot d \cdot n$)

(b) C_s versus ϵ for Direct Request ($C_s = p$) and Sparse-PIR ($C_s = d$)

(c) C_p versus ϵ for AS-Direct Request ($C_p = p$) and AS-Sparse-PIR ($C_p = \theta \cdot d \cdot n$)

(d) C_s versus ϵ for AS-Direct Request ($C_s = p$) and AS-Sparse-PIR ($C_s = d$)

Fig. 6. Parameterized plots for Direct Request and Sparse-PIR, AS-Direct Request, and AS-Sparse-PIR, for $d = 10^2$, $d_a = \frac{d}{2}$, $n = 10^6$, and $u = 10^3$. The dots in the figures show the varying parameter p (for the Direct Request schemes) or θ (for the Sparse-PIR schemes).

target users, the system can be reduced to one not using an anonymity system (choosing $u - 1 = 0$ in the composition lemma leads to $\epsilon_2 = 2 \cdot \epsilon_1$ because of the bound looseness).

Without using a central entity, solutions to counter Sybil attacks are limited, especially when maintaining anonymity. Usual techniques to counter Sybil attacks could be adapted, such as admission control by verifying external identifiers in order to submit requests, requiring a proof of work to increase the cost of Sybil attacks, or deploying social-graph-based Sybil detection.

7 Conclusions

We show that ϵ-private PIR can be instantiated by a number of systems, using dummy queries, anonymous channels, and variants of the classic Chor protocol. Yet some popular naive designs based on dummies or anonymous channels alone fail to provide even this weaker notion of privacy. We argue that the weaker protection provided by ϵ-private PIR may be sufficient to provide some privacy in systems that are so large in terms of database size, but also so popular, that current IT-PIR techniques are impossible to apply. With a large fraction of honest servers even weak (but still ϵ-private) variants of PIR, such as Sparse-PIR, provide near-perfect privacy. Showing that a system is ϵ-private enables smooth composition with an anonymity system, which guarantees that any anonymized ϵ-private PIR mechanism becomes near perfect given a large enough anonymity set.

Acknowledgements

We thank our shepherd Nicholas Hopper and the anonymous reviewers for their helpful feedback in improving this paper. Goldberg thanks NSERC for grant RGPIN-341529; Danezis was supported by H2020 PANORAMIX Grant (ref. 653497) and EPSRC Grant EP/M013286/1; and Toledo by Microsoft Research.

References

[1] Aguilar-Melchor, C., Barrier, J., Fousse, L., Killijian, M.O.: XPIR: Private Information Retrieval for Everyone. Proceedings on Privacy Enhancing Technologies 2016(2), 155–174 (2016)

[2] Aguilar Melchor, C., Gaborit, P.: A Lattice-Based Computationally-Efficient Private Information Retrieval Protocol. In: Western European Workshop on Research in Cryptology (2007)

[3] Andres, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. pp. 901–914 (2013)

[4] Asonov, D., Freytag, J.C.: Almost optimal private information retrieval. In: Dingledine, R., Syverson, P.F. (eds.) Privacy Enhancing Technologies, Second International Workshop, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2482, pp. 209–223. Springer (2002), http://dx.doi.org/10.1007/3-540-36467-6_16

[5] Balsa, E., Troncoso, C., Diaz, C.: OB-PWS: Obfuscation-Based Private Web Search. In: Security and Privacy (SP), 2012 IEEE Symposium on. pp. 491–505. IEEE (2012)

[6] Beimel, A., Stahl, Y.: Robust Information-Theoretic Private Information Retrieval. In: 3rd Conference on Security in Communication Networks. pp. 326–341 (2002)

[7] Berthold, O., Pfitzmann, A., Standtke, R.: The disadvantages of free mix routes and how to overcome them. In: Designing Privacy Enhancing Technologies. pp. 30–45. Springer (2001)

[8] Borisov, N., Danezis, G., Goldberg, I.: DP5: A private presence service. PoPETs 2015(2), 4–24 (2015), http://www.degruyter.com/view/j/popets.2015.2015.issue-2/popets-2015-0008/popets-2015-0008.xml

[9] Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communication of the ACM 24(2) (1981)

[10] Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. Presented at the 36th Annual IEEE Symposium on Foundations of Computer Science (1995)

[11] Danezis, G., Syverson, P.F.: Bridging and fingerprinting: Epistemic attacks on route selection. In: Borisov, N., Goldberg, I. (eds.) Privacy Enhancing Technologies, 8th International Symposium, PETS 2008, Leuven, Belgium, July 23-25, 2008, Proceedings. Lecture Notes in Computer Science, vol. 5134, pp. 151–166. Springer (2008), http://dx.doi.org/10.1007/978-3-540-70630-4_10

[12] Demmler, D., Herzberg, A., Schneider, T.: RAID-PIR: Practical Multi-Server PIR. In: 6th ACM Workshop on Cloud Computing Security (CCSW). pp. 45–56 (2014)

[13] Dingledine, R., Mathewson, N., Syverson, P.F.: Tor: The second-generation onion router. In: USENIX Security Symposium, August 9-13, 2004, USA. pp. 303–320 (2004)

[14] Dwork, C.: Differential privacy. International Colloquium on Automata, Languages and Programming (2006)

[15] Ghinita, G., Kalnis, P., Skiadopoulos, S.: Privé: Anonymous Location-Based Queries in Distributed Mobile Systems. In: 16th International Conference on World Wide Web. pp. 371–380. ACM (2007)

[16] Goldberg, I.: Improving the Robustness of Private Information Retrieval. In: 28th IEEE Symposium on Security and Privacy. pp. 131– 148 (2007)

[17] Henry, R., Huang, Y., Goldberg, I.: One (Block) Size Fits All: PIR and SPIR Over Arbitrary-Length Records via Multi-block PIR Queries. In: 20th Network and Distributed System Security Symposium (2013)

[18] Herrmann, D., Maaß, M., Federrath, H.: Evaluating the security of a DNS query obfuscation scheme for private web surfing. In: Cuppens-Boulahia, N., Cuppens, F., Jajodia, S., Kalam, A.A.E., Sans, T. (eds.) ICT Systems Security and Privacy Protection - 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2-4, 2014. Proceedings. IFIP Advances in Information and Communication Technology, vol. 428, pp. 205–219. Springer (2014), http://dx.doi.org/10.1007/978-3-642-55415-5_17

[19] Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: 2nd international conference on Mobile systems, applications, and services. pp. 177–189 (2004)

[20] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Batch codes and their applications. Proceedings of the 36th Annual ACM Symposium on Theory of Computing (2004)

[21] Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Pervasive Services, 2005. ICPS'05. pp. 88–97. IEEE (2005)

[22] Laurie, B., Langley, A., Kasper, E.: Certificate transparency. RFC 6962 (June 2013)

[23] Lueks, W., Goldberg, I.: Sublinear Scaling for Multi-Client Private Information Retrieval. In: 19th International Conference on Financial Cryptography and Data Security (2015)

[24] Mittal, P., Olumofin, F., Troncoso, C., Borisov, N., Goldberg, I.: PIR-Tor: Scalable Anonymous Communication Using Private Information Retrieval. In: 20th USENIX Security Symposium. pp. 475–490 (2011)

[25] Olumofin, F., Goldberg, I.: Revisiting the Computational Practicality of Private Information Retrieval. In: 15th International Conference on Financial Cryptography and Data Security. pp. 158–172 (2011)

[26] Ostrovsky, R., Kushilevitz, E.: Replication is not needed: single database, computationally-private information retrieval. Proceedings of the 38th Annual Symposium on Foundations of Computer Science (1997)

[27] Roy, I., Setty, S.T.V., Kilzer, A., Shmatikov, V., Wichel, E.: Airavat: Security and privacy for mapreduce. Symposium on

Networked Systems Design and Implementation (2010)
[28] Saint-Jean, F., Johnson, A., Boneh, D., Feigenbaum, J.: Private web search. In: Ning, P., Yu, T. (eds.) Proceedings of the 2007 ACM Workshop on Privacy in the Electronic Society, WPES 2007, Alexandria, VA, USA, October 29, 2007. pp. 84–90. ACM (2007), http://doi.acm.org/10.1145/1314333.1314351
[29] Sarwate, D.: http://math.stackexchange.com/questions/82841/probability-that-a-n-frac12-binomial-random-variable-is-even (2011)
[30] Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. ACM Conference on Computer and Communications Security (2015)
[31] Sion, R., Carbunar, B.: On the practicality of private information retrieval. Proceedings of the Network and Distributed System Security Symposium (2007)
[32] Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In Proceedings of the 23rd UNESIX Security Symposium (2014)
[33] Zhao, F., Hori, Y., Sakurai, K.: Analysis of privacy disclosure in DNS query. In: 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE 2007), 26-28 April 2007, Seoul, Korea. pp. 952–957. IEEE Computer Society (2007), http://dx.doi.org/10.1109/MUE.2007.84

A Proofs of Theorems

A.1 Proof of Composing Naive Mechanisms

Proof. We want to prove in our indistinguishability game scenario that the probability the adversary observes exactly u queries Q_i is bounded above by $\delta_u \leq \left(\frac{p-1}{n-1}\right)^{u-1}$ while the probability they receive no Q_i queries is bounded above by $\delta_0 \leq \left(\frac{n-p}{n-1}\right)^{u-1}$. We first assume that the probability a user chooses one of the two queries (Q_i) given by the adversary is $\Pr_T$.

The probability a non-target user $\mathcal{U}_k$ selects this very Q_i out of his $p-1$ randomly selected requests (the p^{th} one being the adversarially provided non-target query Q_{n_k}) is $\frac{\binom{n-2}{p-2}}{\binom{n-1}{p-1}} = \frac{p-1}{n-1}$ as each record can only be requested once by any given user. As each user is independent, the probability all the users select Q_i is the product of the probabilities, we thus have $\delta_u = \Pr_T \left(\frac{p-1}{n-1}\right)^{u-1}$. Similarly, the probability a non-target user does not select this very Q_i out of his $p-1$ randomly selected requests is $\frac{\binom{n-2}{p-1}}{\binom{n-1}{p-1}} = \frac{n-p}{n-1}$. As each user is independent, the probability none of the users selects Q_i is the product of the probabilities, so $\delta_0 = \Pr_T \left(\frac{n-p}{n-1}\right)^{u-1} \leq \left(\frac{n-p}{n-1}\right)^{u-1}$, and similarly for δ_u. □

A.2 Proof of Security Theorem 1 (Direct Requests)

Proof. We want to prove the following result.

$$\mathcal{L} = \frac{\mathcal{P}_i}{\mathcal{P}_j} = \frac{\Pr(Observation \mid Q_{Target} = Q_i)}{\Pr(Observation \mid Q_{Target} = Q_j)} \leq \frac{1}{d-d_a} \cdot \left(d \cdot \frac{n-1}{p-1} - d_a\right)$$

We first note that the best observation for the adversary is to see exactly one of the adversarially provided target requests, for instance Q_i.

In the first case, the adversary supposes Q_i was sent. Q_j may also have been sent, but in this case a non-colluding database would have received it.

$$\begin{aligned}\mathcal{P}_i &= \frac{d_a}{d} \cdot \binom{n-1}{p-1}^{-1} \cdot \left[\binom{n-2}{p-1} + \frac{d-d_a}{d} \cdot \binom{n-2}{p-2}\right] \\ &= \frac{d_a}{d} \cdot \binom{n-1}{p-1}^{-1} \cdot \left[\binom{n-1}{p-1} - \frac{d_a}{d} \cdot \binom{n-2}{p-2}\right] \\ &= \frac{d_a}{d} \cdot \left[1 - \frac{d_a}{d} \cdot \frac{p-1}{n-1}\right]\end{aligned}$$

In the second case, the adversary supposes Q_j was sent however she only sees Q_i. Q_j must thus have been received by a non-colluding database.

$$\begin{aligned}\mathcal{P}_j &= \frac{d_a}{d} \cdot \frac{d-d_a}{d} \cdot \binom{n-2}{p-2} \cdot \binom{n-1}{p-1}^{-1} \\ &= \frac{d_a}{d} \cdot \frac{d-d_a}{d} \cdot \frac{p-1}{n-1}\end{aligned}$$

Therefore we obtain the result:

$$\begin{aligned}\mathcal{L} = \frac{\mathcal{P}_i}{\mathcal{P}_j} &\leq \frac{\frac{d_a}{d} \cdot \left[1 - \frac{d_a}{d} \cdot \frac{p-1}{n-1}\right]}{\frac{d_a}{d} \cdot \frac{d-d_a}{d} \cdot \frac{p-1}{n-1}} \\ &\leq \frac{d}{d-d_a} \cdot \frac{n-1}{p-1} - \frac{d_a}{d-d_a} \\ &\leq \frac{1}{d-d_a} \cdot \left(d \cdot \frac{n-1}{p-1} - d_a\right)\end{aligned}$$

This concludes the proof. □

A.3 Proof of Security Theorem 2 (Sparse-PIR)

Proof. We represent the p requests sent by the user by $\{0,1\}^{1\times n}$ vectors listed in a $d \times n$ matrix, each column representing a record and each row a request. The adversary $\mathcal{A}$

controlling only a set of the databases will only see some of the rows. $\mathcal{A}$ is interested in the number of ones in the columns, these numbers representing how many times each record has been requested.

We first note that the probability an (d, θ)-Binomial variable is even is $\frac{1}{2} + \frac{1}{2}(1-2\theta)^d$. [29]

The adversary observes only the part of each column v_i corresponding to the corrupt servers d_a. We call the adversary observation for column i, o_i, and the hidden part of the vector h_i. Without loss of generality we consider that $v_i \leftarrow o_i|h_i$ namely that the column for entry i is the concatenation of the observed and the hidden part of the column.

We denote the event the user queried for record α as Q_α. For such a query our mechanism would set the column α, namely v_α, to have odd Hamming weight, and all other column $v_\beta, \beta \neq \alpha$ to have even Hamming weight.

To prove that the mechanism is differentially private we need to show that:

$$\frac{\Pr[\forall i. o_i|Q_\alpha]}{\Pr[\forall i. o_i|Q_\beta]} \leq e^\epsilon$$

However, each column of the query is sampled independently of all others, and thus it suffices to prove that:

$$\frac{\prod_{\forall i.} \Pr[o_i|Q_\alpha]}{\prod_{\forall i.} \Pr[o_i|Q_\beta]} \leq e^\epsilon$$

Since $\Pr[o_i|Q_\alpha]/\Pr[o_i|Q_\beta] = 1$ for $i \notin \{\alpha, \beta\}$, this expression simplifies to:

$$\frac{\Pr[o_\alpha|Q_\alpha] \cdot \Pr[o_\beta|Q_\alpha]}{\Pr[o_\alpha|Q_\beta] \cdot \Pr[o_\beta|Q_\beta]} \leq e^\epsilon$$

We have the following cases depending on the observed parity of o_i, based on the expected parity of the full, and partly unobserved, v_i and v_j:

$$\begin{aligned}
\Pr[o_i \text{ odd}|Q_i] &= \Pr[h_i \text{ even}] \\
\Pr[o_i \text{ even}|Q_i] &= \Pr[h_i \text{ odd}] = 1 - \Pr[h_i \text{ even}] \\
\Pr[o_j \text{ odd}|Q_i] &= \Pr[h_j \text{ odd}] = 1 - \Pr[h_j \text{ even}] \\
\Pr[o_j \text{ even}|Q_i] &= \Pr[h_j \text{ even}]
\end{aligned}$$

For $\theta < 1/2$, it is the case that $\Pr[h_i \text{ even}] > \Pr[h_i \text{ odd}]$ and the differential privacy bound is minimized for:

$$\begin{aligned}
\frac{\Pr[o_\alpha \text{ odd}|Q_\alpha] \cdot \Pr[o_\beta \text{ even}|Q_\alpha]}{\Pr[o_\alpha \text{ odd}|Q_\beta] \cdot \Pr[o_\beta \text{ even}|Q_\beta]} &= \\
\frac{\Pr[h_\alpha \text{ even}] \cdot \Pr[h_\beta \text{ even}]}{\Pr[h_\alpha \text{ odd}] \cdot \Pr[h_\beta \text{ odd}]} &= \\
\frac{\Pr[h_\alpha \text{ even}]^2}{\Pr[h_\alpha \text{ odd}]^2} &= \\
\left(\frac{1/2 + 1/2(1-2\theta)^{|h_i|}}{1 - (1/2 + 1/2(1-2\theta)^{|h_i|})}\right)^2 &= \\
\left(\frac{1 + (1-2\theta)^{|h_i|}}{1 - (1-2\theta)^{|h_i|}}\right)^2 &
\end{aligned}$$

The value of ϵ such that this expression is bounded above by e^ϵ can be expressed in terms of an inverse hyperbolic tangent ($\operatorname{arctanh} x = \frac{1}{2}\ln\left(\frac{1+x}{1-x}\right); |x| < 1$):

$$\epsilon = 4 \cdot \operatorname{arctanh}(1-2\theta)^{|h_i|}$$

This concludes the proof and the upper bound is tight. □

A.4 Proof of the Composition Lemma

Proof. We consider the observations $\mathcal{O}_0 \ldots \mathcal{O}_{u-1}$ as originating from the ϵ_1-private PIR mechanism used by users $\mathcal{U}_0$ to $\mathcal{U}_{u-1}$ respectively. Without loss of generality we consider the target user $\mathcal{U}_t$ is $\mathcal{U}_0$. We try to determine a bound on the following quantity to prove ϵ-privacy:

$$\frac{\Pr(\mathcal{O}_0 \ldots \mathcal{O}_{u-1}|Q_i, Q_{n_1} \ldots Q_{n_{u-1}})}{\Pr(\mathcal{O}_0 \ldots \mathcal{O}_{u-1}|Q_j, Q_{n_1} \ldots Q_{n_{u-1}})} \leq e^{\epsilon_2}$$

However, due to the use of the anonymity system the adversary has a uniform belief about the matching of all observations to all queries, out of the $u!$ possible matchings. Thus we have that:

$$\begin{aligned}
\Pr(\mathcal{O}_0 \ldots \mathcal{O}_{u-1}|Q_x, Q_{n_1} \ldots Q_{n_{u-1}}) = \\
&= \frac{1}{u!} \sum_{i=0}^{u-1} (u-1)! \Pr(\mathcal{O}_i|Q_x) \prod_{j \neq i} \Pr(\mathcal{O}_j|Q_{n_j}) \\
&= \frac{1}{u} \sum_{i=0}^{u-1} \Pr(\mathcal{O}_i|Q_x) \prod_{j \neq i} \Pr(\mathcal{O}_j|Q_{n_j})
\end{aligned}$$

The quantity to be bounded can therefore be re-written as:

$$\frac{\frac{1}{u}\sum_{i=0}^{u-1} \Pr(\mathcal{O}_i|Q_x) \prod_{j \neq i} \Pr(\mathcal{O}_j|Q_{n_j})}{\frac{1}{u}\sum_{i=0}^{u-1} \Pr(\mathcal{O}_i|Q_y) \prod_{j \neq i} \Pr(\mathcal{O}_j|Q_{n_j})} =$$

$$\frac{\Pr(\mathcal{O}_0|Q_x) \prod_{j \neq 0} \Pr(\mathcal{O}_j|Q_{n_j}) + \sum_{i \neq 0} \Pr(\mathcal{O}_i|Q_x) \prod_{j \neq i} \Pr(\mathcal{O}_j|Q_{n_j})}{\sum_{i=0}^{u-1} \Pr(\mathcal{O}_i|Q_y) \prod_{j \neq i} \Pr(\mathcal{O}_j|Q_{n_j})}$$

We are now making a simplifying assumption: We consider that $\Pr(\mathcal{O}_i|Q_z) = \mu$ if the observation $\mathcal{O}_i$ was indeed produced by the query Q_z, and ν otherwise, and also $\mu > \nu$. We rely on the law of large numbers for this assumption to approximate reality, and it is not sensitive to adversary inputs. Since the PIR mechanism is ϵ-private we know that $\mu \leq e^{\epsilon_1}\nu$. This simplifying assumption holds for large numbers of u, since products of multiple individual $\Pr(\mathcal{O}_i|Q_z)$ will tend to be products of the average μ and ν.

The quantity to be bounded now reduces to:

$$\begin{aligned}\frac{\mu^2\mu^{u-2}+(u-1)\nu^2\mu^{u-2}}{\nu^2\mu^{u-2}+(u-1)\nu^2\mu^{u-2}} &= \\ \frac{\mu^2+(u-1)\nu^2}{u\nu^2} &= \\ \frac{\left(\frac{\mu}{\nu}\right)^2+u-1}{u} &\leq \\ \frac{\left(e^{\epsilon_1}\right)^2+u-1}{u} &= \\ e^{\ln(e^{2\epsilon_1}+u-1)-\ln u}\end{aligned}$$

This concludes the proof. □