Repudiative Information Retrieval

Dmitri Asonov*
Humboldt University Berlin
10099 Berlin, Germany
asonov@dbis.informatik.hu-berlin.de

Johann-Christoph Freytag
Humboldt University Berlin
10099 Berlin, Germany
freytag@dbis.informatik.hu-berlin.de

ABSTRACT

Privacy is preserved while retrieving an i-th record from the database of N records if no information is revealed about i, not even to the database server. Repudiation is preserved in the same model if no one can prove, even in cooperation with the server, that a record retrieved is or is not the j-th record for any $1 \leq j \leq N$. The first problem is called PIR, and we call the second problem the RIR problem.

State of the art PIR protocols with optimal query response time and optimal communication require heavy periodical preprocessing. $O(N \log N)$ I/O's are required for preprocessing before answering a query.

In this paper, we reduce preprocessing complexity by weakening PIR to RIR. In particular, we propose a RIR protocol with optimal query response time and communication, and $O(\sqrt{N})$ preprocessing per query.

Categories and Subject Descriptors

F.0 [**Theory of Computation**]: General

General Terms

Algorithms, Security

Keywords

Privacy enhancing technologies, privacy in the digital business, privacy and anonymity in Web transactions

1. INTRODUCTION

The Private Information Retrieval (PIR) problem was initially proposed in [1] and has attracted a lot of attention in the research community [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Given a database of N records, a PIR protocol provides an execution of user queries in such a way, that no information about the content of the queries is revealed, even to the server that has an exclusive access to the database. PIR protocols have a large number of potential applications [12], such as providing privacy of users buying digital goods.

Important characteristics of a PIR protocol, from the user's point of view, are (i) communication complexity between the user and server and (ii) the query response time. Initially proposed PIR protocols possess high complexities in either communication between the server and user [4, 5] or query response time [2, 3, 6, 7, 8].

An effort to achieve optimal both communication and query response time results in $O(N \log N)$ server's preprocessing complexity per query [9, 10, 11]. With some practical assumptions, the server spends several minutes of preprocessing time to be prepared to answer one query [11], which might be intolerable for dynamic business applications.

A natural question arises, whether it is possible to reduce the preprocessing complexity by relaxing the strong privacy requirement from "*no* information about queries is revealed" to "*not much* information about queries is revealed".

By "not much" we mean that, even if some information is revealed, an observer cannot determine for sure if the user queried the 1-st, the 2-nd, the 3-rd,..., or the N-th record. That is, the user can deny any claims of the form "the record you queried is [not] the j-th record", for any j. If a protocol provides the users with this repudiation property, we call it a Repudiative Information Retrieval (RIR) protocol.

Apart from theoretical interest, such protocols would be valuable in any practical scenarios, where, provided that the repudiation property is preserved, users agree to sacrifice some privacy for better performance.

1.1 Our Results

In this paper, we show that if "not much" information should be revealed, $O(\sqrt{N})$ preprocessing complexity is achievable, while keeping optimal communication and query response time. In other words, we construct a RIR protocol with optimal communication and query response time, and $O(\sqrt{N})$ preprocessing complexity per query (in contrast to $O(N \log N)$ preprocessing complexity required for PIR [11]).

Furthermore, we define the robustness of the repudiation property of a protocol, such that a PIR protocol can be seen as a RIR protocol with fully robust repudiation property (Section 2). Then, we show how to construct a RIR protocol with any given robustness of the repudiation property (Section 4). In particular, we observe conditions that turn our RIR protocol into PIR.

*This research was supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316.)

WPES'02, November 21, 2002, Washington, DC, USA

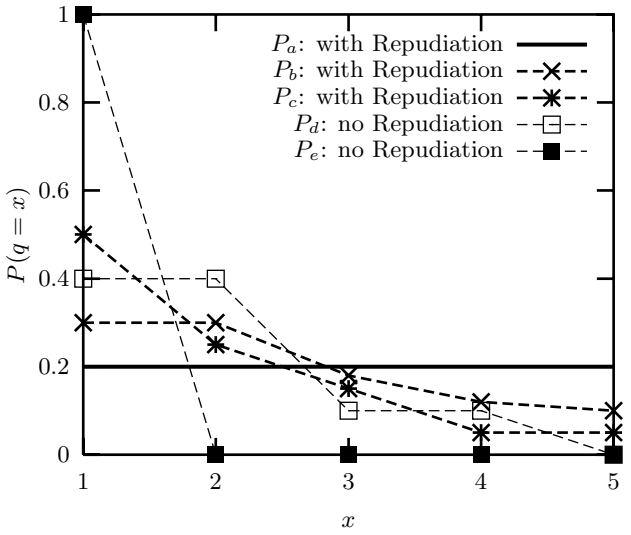

Figure 1: The probability distributions with or without repudiation property.

1.2 Preliminaries and Assumptions

We consider a model, where users are querying a database of N records. A content of the user query is presumed to be of the form "return the i-th record". Under the observer we mean (any conspiracy with) the database server exclusively accessing the database, that tries to figure out the content of the user query. We denote information about queries known to observer by $I_{revealed}$. This information is determined by the protocol used.

2. DEFINING REPUDIATION AND ASSESSING ITS ROBUSTNESS

We define whether the repudiation property is supported or not by a given protocol in Section 2.1. For the cases where the repudiation property is supported, we suggest the assessment of the robustness of this property in Section 2.2.

2.1 Repudiation Property

Let us assume, that the user has run any protocol to execute his query q ="return the i-th record", or $q = i$ for short. We say, that this protocol assures the repudiation property, if q can be of any value between 1 and N for an observer. So that the user can deny any claim of the type "the record you queried is [not] the j-th record". Formally, the repudiation property is preserved iff:

$$0 < P(q = j|I_{revealed}) < 1, \quad \forall j : 1 \leq j \leq N \qquad (1)$$

If the protocol assures the repudiation property, we call it a repudiative information retrieval (RIR) protocol. Examples of probability distributions $P(q = x)$ provided by different hypothetical protocols[1] are shown in Fig. 1 for $N = 5$.

Now that we defined RIR, it is important to understand clearly the relationship between RIR, PIR, and just a retrieval of the required record without any privacy techniques. The former we call a download for short. All the three are compared in Table 1 based on distributions $P(q|I_{revealed})$ they produce. Distribution P_e belongs to a download protocol because all information about query q is revealed, i.e., Shannon's entropy $H(P_e) = 0$. Distributions P_a, P_b, P_c, P_d represent protocols that hide some information about query. P_a, P_b, and P_c are produced by RIR protocols (labeled as RIR_N in the table) and so the user can decline any claims "the record you queried is [not] the j-th record" for any $1 \leq j \leq N$. Note that distribution P_a, although representing a RIR protocol, can also be classified for a PIR protocol because no information about query is revealed in this distribution. Distribution P_d does not satisfy 1 and thus a protocol with such a distribution is not a RIR protocol.[2]

[1]The probabilities in the distributions on the figure are sorted (in the descending order) to make it easier to compare the possible patterns of distributions.

Table 1: Possible probability distributions classified.

Protocol	Parameter			
	distribution $P(q)$	repudiation assured	preprocess. complexity	info kept private
Download	P_e	no	0	$H(P) = 0$
PIR_N	P_a	yes	$O(NlogN)$	$H(P)$ =max
RIR_N	P_a, P_b, P_c	yes	?	0< H ≤max
$RIR_{n<N}$	P_d	no	?	0< H <max

2.2 Assessing the Robustness of Repudiation

As we can see from Fig. 1, protocols providing repudiation property may have different distributions $P(q = x)$. For example, for one RIR protocol the user query can be guessed with probability 0.5, and for another protocol with probability of $1/N = 0.2$ as if no information about the query were revealed. Formally, claims about the user query can be true with different probabilities for different RIR protocols. This observation can be interpreted as if repudiation property provided can be of different quality or robustness[3].

We construct different RIR protocols in the next sections. In order to differentiate those protocols by the quality of repudiation provided, we may need an assessment for robustness of repudiation property.

2.2.1 Distilling the Criteria for Assessment Function

First, we agree on the minimal value for the robustness of repudiation (RR): We say that $RR = 0$ iff repudiation (as defined in the previous section) is not preserved. Second, we say that the robustness of repudiation is maximal ($RR = 1$), iff no information about the query is revealed. Thus, a RIR protocol with $RR = 1$ is a PIR protocol. For example, for probability distributions shown in Fig.1 we presume:

$$RR(P_a) = 1; \quad RR(P_d) = 0; \quad RR(P_e) = 0$$

$$0 < RR(P_b) < 1; \qquad 0 < RR(P_c) < 1$$

Third, we have to decide on how to assess $RR(P)$ in the open interval $]0, 1[$. For instance, we must decide on values for $RR(P_b)$ and $RR(P_c)$. This decision determines, for example, which of the two protocols (corresponding to distributions

[2]However, such a distribution could belong to a RIR protocol if the number of records in the database is reduced from 5 to 4. We denote protocols producing such distributions as $RIR_{n<N}$ and do not consider in this paper.

[3]The meaning of the word robustness in statistical analysis is different from one in plain English. Please, do not be confused: We use the word robustness as in plain English; it can be substituted with synonym like quality.

P_b and P_c) provides more robust repudiation. Our approach to assessing $RR(P)$ in the interval $]0,1[$ is as follows.

For any distribution P, we show how to obtain it by morphing incrementally the perfect distribution P_a that corresponds to $RR = 1$. Namely, the morphing is performed through a number of elementary changes. An elementary change is made by changing only two points in distribution by adding some δ to one probability and subtracting δ from another.

Intuitively, a good assessment function $RR(P)$ must monotonically decrease as distribution is being morphed from P_a to P, because with each new change, the distribution moves further away from the perfect distribution (Fig. 2).

In summary, we postulate that the following three properties are necessary and sufficient for a function (RR) to serve as an assessment for the robustness of repudiation of a protocol:

1. $RR(P) = 0 \quad iff \quad \exists i: \quad p_i = 0 \quad (or \quad p_i = 1)$
2. $RR(P) = 1 \quad iff \quad \forall i: \quad p_i = \frac{1}{N}$
3. $\forall P = \{p_1, p_2, .., p_N\}, P' = \{p_1 + \delta, p_2 - \delta, .., p_N\}$: $RR(P) > RR(P') \quad if \quad p_1 \geq p_2$

For example that Fig. 2 demonstrates, the third property implies that for any function $RR(P)$ eligible for assessing robustness of repudiation must hold:

$$RR(P_a) > RR(P'_a) > RR(P''_a) > RR(P_b)$$

The task now is to find a function satisfying all the three criteria.

2.2.2 Function Satisfying The Criteria.

An example of a function satisfying all the three criteria is:

$$RR(P) \underset{def}{=} N^2 \frac{1}{\sum_{1 \leq i \leq N} \frac{1}{p_i}} \tag{2}$$

Other functions satisfying the three criteria include, for instance:

$$RR(P) = N^N \prod_{1 \leq i \leq N} p_i \tag{3}$$

$$RR(P) = \frac{1}{(\frac{1}{N}(1 - \frac{1}{N}))^N} \prod_{1 \leq i \leq N} p_i(1 - p_i) \approx \approx eN^N \prod_{1 \leq i \leq N} p_i(1 - p_i) \tag{4}$$

$$RR(P) = \frac{N \frac{1}{\frac{1}{N}(1 - \frac{1}{N})}}{\sum_{1 \leq i \leq N} \frac{1}{p_i(1-p_i)}} = \frac{\frac{N^3}{N-1}}{\sum_{1 \leq i \leq N} \frac{1}{p_i(1-p_i)}} \tag{5}$$

Although any of the functions mentioned can be used to assess robustness of repudiation, we prefer alternative 2 based on its simplicity.

3. BASIC REPUDIATIVE INFORMATION RETRIEVAL PROTOCOL

In this section we present an example of a RIR protocol. It provides a particular robustness of repudiation, namely $RR = O(\frac{1}{N})$. In the next section we will extend our protocol for providing any given repudiation robustness.

Similar to the previous PIR protocols [6, 7, 9, 10], our RIR protocol uses a notion of secure coprocessor (SC). From a theoretical point of view, the notion of SC can be substituted with the notion of the third party that (i) runs the protocol certified by users and the server and cannot alter it and (ii) discloses nothing above the protocol specifies. Consequently, the data processed inside the SC cannot be observed from outside of the SC. For a good starting point, we sketched a simple PIR protocol based on SC in Appendix A.

Algorithm 1 that runs inside a SC (Fig. 4) provides a RIR protocol. Before starting the protocol, the SC must prepare database RS to use it later as an input for Algorithm 1. Each record in RS is randomly selected from the original database. Each record is also kept encrypted with the private key of the SC, so that no one can observe its identity. Algorithm 1 takes one unused record from RS to answer one query. Thus the database RS must be renewed periodically.

Input: The database of N records $DB[1, .., N]$; a database of randomly selected (encrypted) records $RS[1, ...]$; a number of previously answered queries k; a record's number to retrieve i;
Output: $DB[i]$: i-th record of the database DB;

```
1: read_into_SC(Result ⇐ decrypt(RS[k + 1]))
2: if index(RS[k + 1]) == i then
3:    Range = {1, ..., N} \ {i}
4:    h = select_random_from(Range)      {Select randomly
                  one of the records referred in Range}
5:    read_into_SC(Temp ⇐ DB[h])
6: else
7:    read_into_SC(Result ⇐ DB[i])
8: end if
9: k = k + 1
10: output_from_SC(encrypt(Result))      {Encrypt and
                  output Result from the SC}
```

Algorithm 1: An example of a RIR protocol.

In the rest of this section we (i) analyze the robustness of repudiation of the proposed protocol, (ii) consider the case for multiple queries, and (iii) discuss the complexity of preparing x records for the database RS.

3.1 Analyzing Robustness of the Protocol

The basic idea behind the algorithm is as follows. As the query arrives into the SC, the SC decrypts the query and obtains i – the number of a record desired by the user. Then the SC reads the next unused record from RS, decrypts it, and checks if this is the record requested by the user. If yes, the SC reads any dummy record from the original database DB, but sends the read RS record as a response, encrypted with the user key. If no, the SC reads the desired record directly from DB, and sends it as a response, encrypted. The former and the latter outcomes have probabilities of $1/N$ and $(N-1)/N$ correspondingly:

$$P(q = index(RS[k+1])) = \frac{1}{N} = P(q \neq d) \tag{6}$$

$$P(q = d) = \frac{N-1}{N} \tag{7}$$

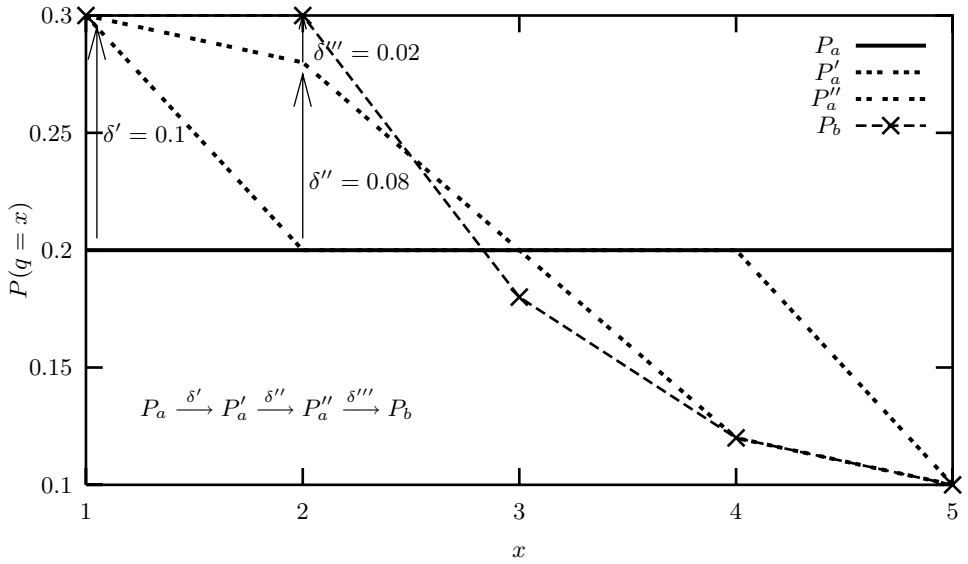

Figure 2: The distribution P_a is morphed into P_b through 3 elementary changes. Each change increases the difference between the current distribution and P_a.

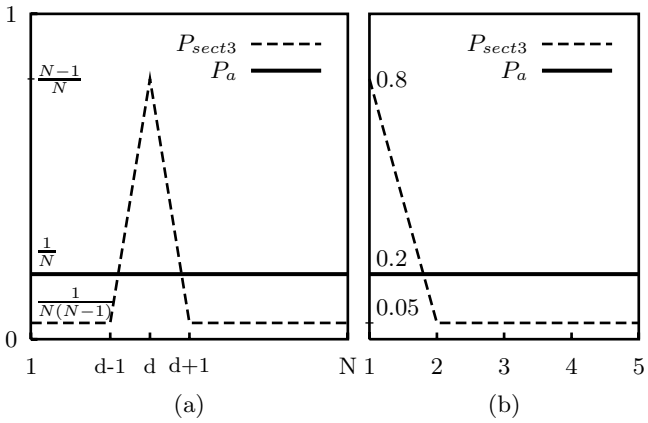

Figure 3: (a) An example of probability distribution corresponding to proposed RIR protocol. (b) The distribution pattern for $N = 5$ (with ordered probabilities).

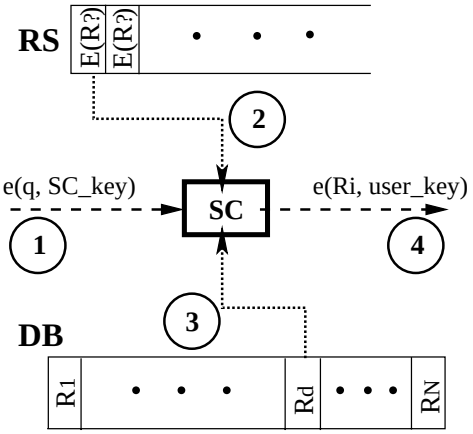

Figure 4: A scheme of a RIR protocol.

where function $index(RS[k+1])$ denotes the number of the record from the original database DB that $RS[k+1]$ represents in an encrypted form; d denotes the record number read by the SC from DB to answer a query.

The probability distribution $P_{sect3}(q = x)$ provided by our protocol is shown in Fig.3. This protocol is repudiative, i.e., equation 1 holds because an observer can neither reject nor prove that the record of interest to the user was record number j, for all $1 \leq j \leq N$.

Not surprisingly, robustness of repudiation provided by our protocol is less than perfect:

$$RR(P_{sect3}) = \frac{N^2}{\sum\limits_{1 \leq i \leq N} \frac{1}{p_i}} = \frac{N^2}{N(N-1)^2 + \frac{N}{N-1}} = \\ = O\left(\frac{1}{N}\right) < RR(P_a) = 1 \quad (8)$$

3.2 Multiple Queries

The amount of information revealed remains zero for PIR (and for RIR protocols with full robustness of repudiation) as the number of queries answered grows. Several questions arise when considering a RIR protocol with $RR < 1$ executing several queries, because the amount of information revealed is not zero and grows with the number of queries answered. Namely, if a user queries a database several times using a RIR protocol then:

1. Does the form of probability distribution $P(q = x)$ change from query to query?
2. Does the probability of a certain record (number s) to be accessed by at least one of the queries increase with the number of queries answered? If yes, does it reach 1 for some number of queries?

The answer is no for the first question, because the queries

are executed independently from each other. That is, independently from the number of queries answered, the distribution $P(q = x)$ is of the same form for all queries (and is due to Figure 3a for the protocol presented above). Thus, the repudiation property holds for every query answered.

To answer the second question, assume k queries are answered. Formally, we are looking for the probability:

$$p_{s,k} \underset{def}{=} p(q_1 = s, \, or \, q_2 = s, \ldots, \, or \, q_k = s) = 1 - P(q_1 \neq s, \, and \, q_2 \neq s, \ldots, \, and \, q_k \neq s) = 1 - P(q_1 \neq s) \cdot P(q_2 \neq s) \cdot \ldots \cdot P(q_k \neq s) \quad (9)$$

It can be shown that this probability grows with k (even if no information revealed). However, $p_{s,k}$ never reaches 1. As a consequence, the robustness of repudiation associated with the distribution $(p_{s,k}, 1 - p_{s,k})$ never reaches 0 even if all user queries are equal. For example, in case of PIR it holds for $p_{s,k}$:

$$p_{s,k}^{PIR} = 1 - (1 - 1/N)^k = 1 - \left(\frac{N-1}{N}\right)^k, \quad \forall s$$

For another example, we consider the RIR protocol described in this section. Let also assume that for all k queries the SC reads the record number d' from the open database. Then, with the help of equation 9 we obtain:

$$p_{s,k} = 1 - (1 - (N-1)/N)^k = 1 - \frac{1}{N^k}, \quad for \; s = d'$$

$$p_{s,k} = 1 - (1 - 1/(N(N-1))) \approx 1 - \left(\frac{N^2-1}{N^2}\right)^k, \; \forall \, s \neq d'$$

Please note that conclusions made in this subsection hold also for the RIR protocol presented in the next section, as well as for any other RIR protocol.

3.3 Complexity of Preprocessing

We discuss briefly the complexity of preparing one record for the database RS. For simplicity, we assume that internal memory of a SC is large enough for storing O(1) database records. A straightforward approach is to read the entire database through the SC, but select one random record to keep inside the SC and, after the entire database is read through, output the encrypted record and store it in RS. Since the SC must read N database records through, the upper bound for the I/O complexity of preparing one record for RS is $O(N)$. In [11] the algorithm with $O(\sqrt{N})$ I/O complexity is developed. We will rely on this result within our paper.

3.4 Summary

We constructed a RIR protocol with both query response time and the communication of $O(1)$ complexity. The preprocessing complexity is $\sqrt{N}$ per query compared to $NlogN$ for PIR with the same query response time and communication [10, 11]. However, users might be unsatisfied with the low robustness provided by the protocol above. Now we want to extend our RIR protocol to provide any specific robustness of repudiation.

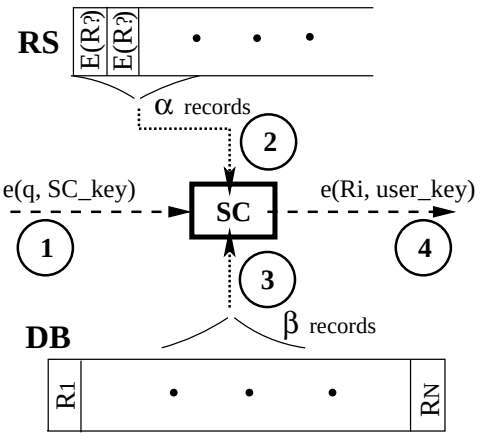

Figure 5: A RIR protocol with configurable robustness of repudiation.

4. VARYING ROBUSTNESS OF THE RIR PROTOCOL

In this section we consider the problem of constructing RIR protocols with a given robustness between 0 and 1. Namely, in the first part of this section we present a RIR protocol with two parameters. Next, we show that by varying this parameters the robustness between 0 and 1 can be achieved. In particular, we show for which parameters this protocol gains full robustness, thus transmuting in a PIR protocol.

4.1 A Parameterized RIR Protocol

We present Algorithm 2 that extends Algorithm 1 (see also Fig. 5). The only difference between the two is in that in the extended version the SC reads α records from RS database and β records $(DB[d_1], ..., DB[d_\beta])$ from DB database, whereas $\alpha = \beta = 1$ for the basic protocol presented earlier. The query response time of the protocol is $O(\alpha + \beta)$ and the preprocessing complexity per query is $O(\alpha\sqrt{N})$.

4.2 How Parameters Determine Robustness of Repudiation

Our goal is to see how the robustness of repudiation depends on α and β parameters of the protocol. In order to find robustness of repudiation of the protocol we first calculate the probability distribution $P(q = x)$.

The probability of the query being not among $d_1, ..., d_\beta$ is equal to the probability of finding the required record among the α records read from RS:

$$P(q \notin \{d_1, ..., d_\beta\}) = P(q \in \{index(RS[k{+}1, ..., k{+}\alpha])\}) = 1 - \left(\frac{N-1}{N}\right)^\alpha \quad (10)$$

The probability of the query being among $d_1, ..., d_\beta$ is calculated as:

$$P(q \in \{d_1, ..., d_\beta\}) = 1 - P(q \notin \{d_1, ..., d_\beta\}) = \left(\frac{N-1}{N}\right)^\alpha \quad (11)$$

Input: The database of N records $DB[1,..,N]$; a database of randomly selected (encrypted) records $RS[1,...]$; a number of previously answered queries k; a record's number to retrieve i; α and β - parameters of the protocol;
Output: $DB[i]$: i-th record of the database DB;

```
1: GotResult = no
2: for g = 1 to α do
3:    read_into_SC(Temp ⇐ decrypt(RS[k * α + g]))
4:    if index(RS[k + 1]) == i then
5:       Result = Temp            {Copy Temp into Result}
6:       GotResult = yes
7:    end if
8: end for
9: Range = {1, ..., N} \ {i}
10: if GotResult == yes then
11:    H = select_random_from(Range, β) {Select randomly
                β distinguished records referred in Range}
12: else
13:    H = select_random_from(Range, β − 1)   {Select
          randomly β − 1 distinguished records referred in
                                              Range}
14:    H[β] = i                 {Select i for β-th element in H}
15: end if
16: sort(H)
17: for g = 1 to β do
18:    read_into_SC(Temp ⇐ decrypt(DB[H[g]]))
19:    if H[g] == i then
20:       Result = Temp            {Copy Temp into Result}
21:    end if
22: end for
23: k = k + 1
24: output_from_SC(encrypt(Result))          {Encrypt and
                                  output Result from the SC}
```

Algorithm 2: Parametric repudiative information retrieval protocol.

Now we have values for $P(q=x)$ for all $1 \le x \le N$ (Fig. 6a):

$$P(q=x) = \frac{1}{\beta}\left(\frac{N-1}{N}\right)^{\alpha}, \quad \forall x \in \{d_1, ..., d_\beta\} \tag{12}$$

$$P(q=x) = \frac{1}{N-\beta}\left(1 - \left(\frac{N-1}{N}\right)^{\alpha}\right), \forall x \notin \{d_1, ..., d_\beta\} \tag{13}$$

Finally, we can calculate the robustness of repudiation as a function of α and β (also shown in Fig. 7):

$$RR(\alpha,\beta) = \frac{N^2}{\sum\limits_{1\le i\le N} \frac{1}{p_i}} = \frac{N^2}{\frac{(N-\beta)^2}{1-\left(\frac{N-1}{N}\right)^{\alpha}} + \frac{\beta^2}{\left(\frac{N-1}{N}\right)^{\alpha}}} \tag{14}$$

Note that for $\alpha = \beta = 1$ equation 14 is identical to equation 8.

4.3 When Our RIR Protocol Turns into PIR

In this subsection we determine such α and β that our RIR protocol can be seen as a PIR protocol, i.e., when our protocol reveals no information about query. This is the case when and only when distributions $P(q=x)$ and P_a are identical in Fig. 8:

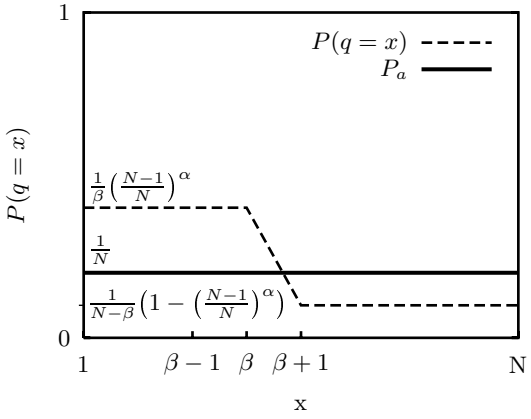

Figure 6: The distribution $P(q=x)$ for given α, β; probabilities are ordered. (Note that this graph looks different for different α, β.)

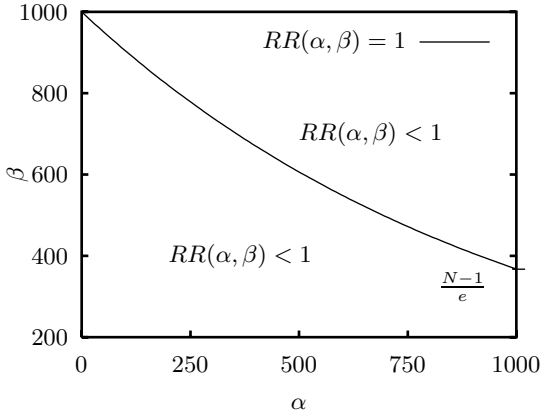

Figure 8: Determining α and β that correspond to a PIR protocol ($N = 1000$).

$$\frac{1}{N} = \frac{1}{\beta}\left(\frac{N-1}{N}\right)^{\alpha} = \frac{1}{N-\beta}\left(1-\left(\frac{N-1}{N}\right)^{\alpha}\right)$$

$$\beta = (N-1)\left(\frac{N-1}{N}\right)^{\alpha-1} \tag{15}$$

Figure 6b exhibits the equation above and shows the dependence between α and β for which our protocol performs like a PIR protocol.

The response time $O(\alpha+\beta)$ of the protocol is equal or higher than $O(N)$. This makes the use of the RIR protocol for PIR inefficient, because response times of the original PIR protocols range from $O(1)$ to $O(N)$ [10, 3, 7].

5. RELATED WORK

So far literature has not reported any work on relaxing privacy requirements for PIR. However, some research exist that can still be related to our work.

5.1 Deniable Encryption

Encryption is deniable [13, 14] if several (or even any) cleartexts can be thought of as a source for a given encrypted

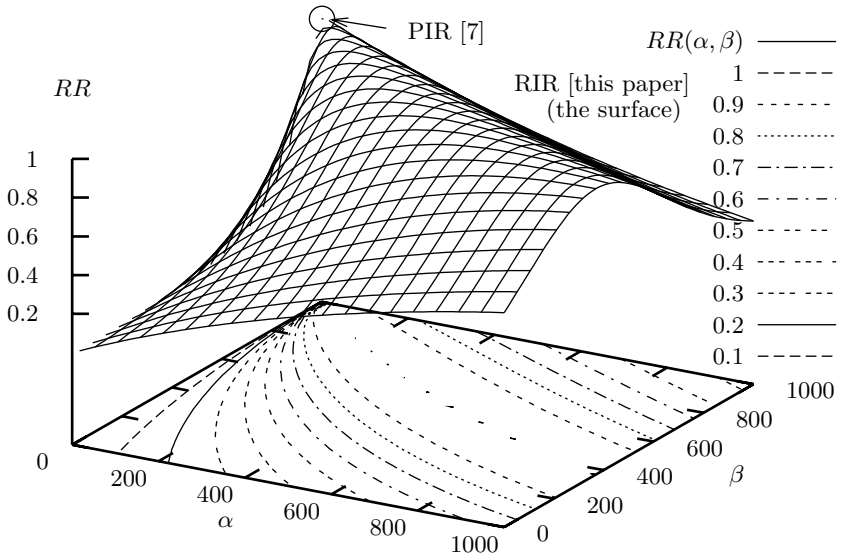

Figure 7: Robustness of repudiation of a RIR protocol is a function of α and β.

message. Repudiative information retrieval is similar to deniable encryption in that any of the database records can be thought of as a target of the user query.

5.2 Alternatives to Quantification of Repudiation

Different formalizations of anonymity were proposed recently [15, 16, 17, 18, 19]. Most of them are designed to compare the quality of different anonymizing networks or services.

We cannot apply these approaches to formalize the robustness of repudiation provided by a RIR protocol because the subjects differ principally. Anonymity approaches aim at hiding identities of the acting users, while RIR (with PIR as its extreme form) aims at hiding the actions performed by users. So we provided an original quantification for the repudiation property of RIR as described above.

An alternative approach to design quantification for the repudiation is to consider measures of distances between probability distributions [20]. Given the probability distribution $P(q = x) = P_a$ of PIR as a standard, one could measure the distances between P_a and any P_{x1} and P_{x2} in oder to compare the robustness of repudiation associated with P_{x1} and P_{x2}.

We examined several known measures of distances and concluded that these measures are inconvenient to use as a repudiation measure for different reasons. The main reason is that RR function based on those measures would not satisfy the three properties defined in Section 2.2.

Finally, we note that Shannon's entropy measure is unsuitable for measuring robustness of repudiation. This is because the entropy of a distribution P with $p_i = 0$ (for some i) is not necessary 0 whereas the robustness of repudiation must be zero in this case, as defined by equation 1.

6. DISCUSSION

This section is mainly motivated by the questions raised by the WPES'02 auditorium on 21st November 2002.

6.1 Redefining Repudiation

We say that the repudiation property is preserved if the observer cannot exclude *any* of the N items from being a possible result of the query. A related question is: What happens if we substitute the word "any" in the previous sentence with "some $z < N$"? In other words, one might consider a more relaxed definition for repudiation, with a certain number z of points in the probability distribution being allowed to be equal 0.

We did not investigate this relaxed definition, because all the conclusions made with our strong definition would repeat themselves for the relaxed case. That is, a RIR in terms of the relaxed definition can be reduced to RIR in terms of the strong definition of repudiation just by substituting N with z within this paper.

6.2 Yet Another Alternative to Quantification of Repudiation

This subsection accomplishes the discussion about alternatives for the definition of robustness of repudiation initiated in Section 5.2.

The alternative definition we discuss here is informal and based on the following observation. The probability distribution corresponding to the full robustness is a plain line (P_a, Fig.1, which is the case of PIR). And an example of the distribution with no robustness of repudiation at all is a curve with a great peak (P_e). So, informally, one could alternatively define the robustness of repudiation using a measure of smoothness of the distribution. A curve with larger and numerous peaks corresponds to the less robust repudiation and vice versa.

Our formal definition supports this observation indeed. The morphing approach we used to define formally the robustness of repudiation is just another interpretation of the described observation.

6.3 Misinforming the Observers

The concept of RIR (PIR) is to reveal only partial (or no) information about the user query. Now let us consider

a kind of generalized concept: Let the user and the SC act so that not only all or part of *true* information about query content is hidden, but also some *false* information about the query content is revealed to confuse the observer [21].

For better understanding, imaging a user intends to query a digital book with some compromising title XXX. If he queries this item with PIR, then no information about the title is revealed. The questions are: (i) Does it make sense (and, (ii) is it possible) to reveal false information about the content of the query to misinform the observers? Misinformation in this context means that it looks like as if an item with a neutral title YYY is retrieved with high probability. We believe that these questions require separate investigation, however, we outline some preliminary answers.

Regarding the first question, it is important to recall that we presume the sources of the protocols loaded into the SC to be publicly known. This means, the observers (such as the server) will be notified if the protocols are modified to misinform the observers. Thus they will not "buy" the misinformation, and will consider any information revealed with scepticism. From this prospective, such misinformation does not make much sense.

Apart from the preliminary answer to the first question, let us consider the second one. The RIR protocol proposed in this paper can be easily modified to support the misinforming feature. Namely, instead of randomly choosing β (or $\beta - 1$) records to read from the plaintext $DB[N]$ database, the SC must choose the records among the items with neutral content (defined by the user) only.

7. CONCLUSION AND FUTURE WORK

Private Information Retrieval aims at retrieving one record of the user's choice from a database of N records in such a way that no one, not even the server, can notify the identity of the record. PIR is known for its heavy query response time or preprocessing complexities.

In order to reduce the preprocessing complexity of the protocol, we relaxed the privacy requirement in that some information about the record identity is allowed to be revealed. However, the information revealed should not be enough to say definitely whether it was the record number 1, or 2, ... or N. So the user is provided with repudiation property.

We constructed such protocols and call them Repudiative Information Retrieval Protocols (RIR). Our RIR construction can be customized according to the robustness of repudiation required. For a small robustness, the preprocessing complexity up to $O(\sqrt{N})$ per query can be achieved compared to $O(NlogN)$ for PIR with the same query response time and communication.

Full robustness of repudiation of a RIR protocol means PIR, and so our RIR protocol can be chosen to serve as a PIR protocol. However, existing PIR protocols have smaller complexities.

PIR protocols exist of two types: based on general purposed hardware and based on a tamper proof device. We demonstrated a RIR protocol constructed with the use of a tamper proof device. An open issue is constructing a practical RIR protocol without use of a tamper proof device.[4]

[4]In fact, constructing a practical PIR protocol without use of a tamper proof device remains an open issue too [10].

Acknowledgements: Anonymous referees have given many valuable comments, one of which provoked a new subsection. We would also like to thank Bernhard Thalheim, Hans-Joachim Lenz, Steffen Jurk, Mattis Neiling, and Peter Rieger for the enormous feedback they provided. This research was supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316).

8. REFERENCES

[1] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proceedings of 36th FOCS*, 1995.

[2] Eyal Kushilevitz and Rafail Ostrovsky. Replication is NOT needed: Single-database computationally private information retrieval. In *Proceedings of 38th FOCS*, 1997.

[3] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *Proceedings of EUROCRYPT'99*, 1999.

[4] Feng Bao, Robert H. Deng, and Peirong Feng. An efficient and practical scheme for privacy protection in the e-commerce of digital goods. In *Proceedings of the 3rd International Conference on Information Security and Cryptology*, December 2000.

[5] Claus Peter Schnorr and Markus Jakobsson. Security of signed elgamal encryption. In *Proceedings of ASIACRYPT'00, LNCS 1976*, December 2000.

[6] Sean W. Smith and Dave Safford. Practical private information retrieval with secure coprocessors. Technical report, IBM Research Division, T.J. Watson Research Center, July 2000.

[7] Sean W. Smith and Dave Safford. Practical server privacy with secure coprocessors. *IBM Systems Journal*, 40(3), September 2001.

[8] Aggelos Kiayias and Moti Yung. Secure games with polynomial expressions. In *Proceedings of 28th ICALP*, 2001.

[9] Dmitri Asonov and Johann-Christoph Freytag. Almost optimal private information retrieval. Technical Report HUB-IB-156, Humboldt University Berlin, November 2001.

[10] Dmitri Asonov and Johann-Christoph Freytag. Almost optimal private information retrieval. In *Proceedings of 2nd Workshop on Privacy Enhancing Technologies (PET2002), San Francisco, USA*, April 2002.

[11] Dmitri Asonov and Johann-Christoph Freytag. Private information retrieval, optimal for users and secure coprocessors. Technical Report HUB-IB-159, Humboldt University Berlin, May 2002.

[12] Dmitri Asonov. Private information retrieval - an overview and current trends. In *Proceedings of the ECDPvA Workshop, Informatik 2001, Vienna, Austria*, September 2001.

[13] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In *Proceedings of Advances in Cryptology, (CRYPTO-97)*, June 1997.

[14] Ronald L. Rivest. Chaffing and winnowing: Confidentiality without encryption. `http://theory.lcs.mit.edu/~rivest/chaffing.txt`, April 1998.

[15] Vitaly Shmatikov and Dominic J.D. Hughes. Defining anonymity and privacy. In *Proceedings of Workshop on Issues in the Theory of Security (WITS '02)*, January 2002.

[16] Vitaly Shmatikov. Probabilistic analysis of anonymity. In *Proceedings of 15th IEEE Computer Security Foundations Workshop (CSFW)*, June 2002.

[17] Paul F. Syverson and Stuart G. Stubblebine. Group principals and the formalization of anonymity. In *Proceedings of World Congress on Formal Methods*, September 1999.

[18] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of 2nd Workshop on Privacy Enhancing Technologies (PET2002), San Francisco, USA*, April 2002.

[19] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Proceedings of 2nd Workshop on Privacy Enhancing Technologies (PET2002), San Francisco, USA*, April 2002.

[20] Alison Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3), December 2002.

[21] Dmitri Asonov and Neil K. Daswani. Pesonal communication, November 2002.

[22] Sean W. Smith, Elaine R. Palmer, and Steve H. Weingart. Using a high-performance, programmable secure coprocessor. In *Proceedings of the 2nd International Conference on Financial Cryptography*, February 1998.

APPENDIX

A. PIR PROTOCOLS WITH SECURE CO-PROCESSOR

Smith et al. [6, 7] make use of a tamper-proof device to implement the following PIR protocol.

A secure coprocessor (a tamper-proof device) is used as a black box, where the selection of the requested record takes place. Although hosted at the server side, the SC is designed so that it prevents any one from accessing its memory from outside [22]. A SC can prove what software is installed inside and whether it was changed in the past.

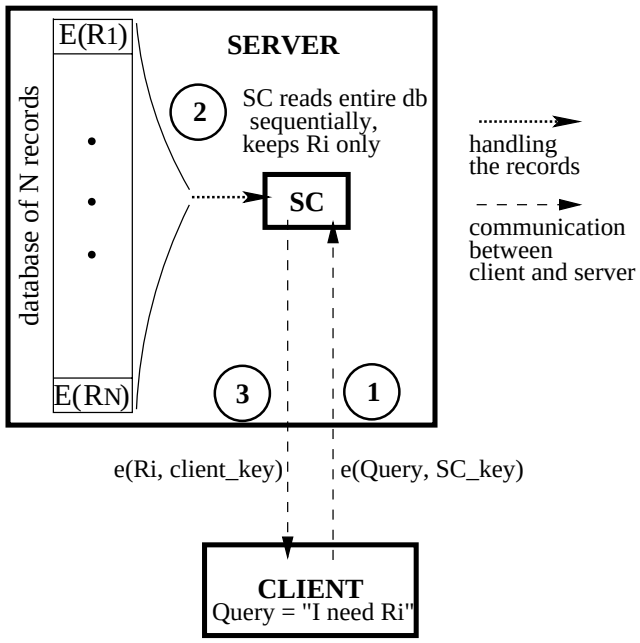

Figure 9: I/O flows in the PIR with SC

The basic protocol runs as shown in Fig. 9. The client encrypts the query "return the i-th record" with a public key of the SC, and sends it to the server. The SC receives the encrypted query, decrypts it, reads through the entire database, but leaves in memory the requested record only. The protocol is finished after the SC encrypts the record and sends it to the client.

To provide integrity, the SC keeps all records of the database encrypted. The main disadvantage of this PIR is the same as that of the PIR described in [2, 3]: $O(N)$ query response time, which is intolerable for large databases.

[9, 10] is another PIR protocol that employs a SC. It manages to reach $O(1)$ response time, although requiring up to $O(NlogN)$ I/O operations per query at preprocessing.