Efficient transformation capabilities of single database private block retrieval

Radhakrishna Bhat¹ · N. R. Sunitha²

Received: 10 August 2021 / Accepted: 2 December 2021 / Published online: 19 January 2022
© The Author(s) 2022

Abstract

Private information retrieval (PIR) is one of the promising techniques to preserve *user privacy* in the presence of *trusted-but-curious* servers. The information theoretically private query construction assures the highest user privacy over curious and unbounded computation servers. Therefore, the need for information-theoretic private retrieval was fulfilled by various schemes in a variety of PIR settings. But, there is a lack of efficient encryption switching scheme which supports efficient switching between information-theoretic to/from computationally bounded PIRs. We propose a combination of new bit connection methods called *rail-shape* and *signal-shape* and new quadratic residuosity assumption based family of trapdoor functions for generic single database private block retrieval (PBR). The main goal of this work is to show that the possibility of mapping from computationally bounded privacy to information theoretic privacy or vice-versa in a single database setting using newly constructed bit connection and trapdoor function combinations. Notably, the proposed schemes are single round, memoryless and plain database schemes (at their basic constructions).

Keywords Private information retrieval · Information-theoretic privacy · User privacy · Private Block Retrieval · Oblivious transfer · Probabilistic encryption

1 Introduction

The goal of any privacy critical applications is to preserve the underlying privacy (like user privacy or server privacy or data privacy) with guaranteed confidentiality primitive (i.e., information theoretic).

Among all other user privacy-preserving techniques, Private Information Retrieval (PIR) is one of the prominent privacy-preserving techniques to preserve both *user privacy* and *data privacy* introduced by Chor et al. [12,13]. The private information retrieval also called as special case of 1-out-of- n oblivious transfer involves two communicating parties: *user* and *server* in which *user* privately reads a sin-

gle bit from *server's* n bit database. The basic goal of Chor et al. [12,13] was to provide the highest confidentiality to the user's interest (maybe index, pattern, graph moves etc.) for real-time privacy applications. Since then, comprehensive research has been carried out in several dimensions of PIR including relaxing the privacy level from information-theoretic to a computationally bounded setting, reducing communication and computation overhead, reducing the number of rounds and number of servers involved, extending to private write etc.

One of the natural extensions to PIR protocol is Private Block Retrieval (PBR) in which *user* privately reads v bit block (instead of a bit) from *server's* u block database. Based on the level of privacy, the PIR protocol is broadly divided into two groups: information-theoretic PIR and computationally bounded PIR as described below.

- *Information-theoretic PIR* (itPIR): If the PIR protocol involves information-theoretically private queries with non-colluding replicated database server entities then such scheme is considered as information-theoretic PIR (itPIR) in which the user privacy is preserved through the information-theoretically private queries. Several

✉ Radhakrishna Bhat
rsb567@gmail.com

N. R. Sunitha
nrsunithasit@gmail.com

¹ Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India

² Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India

information-theoretic schemes [3,4,8,17,19,28,32] and some PBR extensions [2,5,13,14,16,21,30,31,37] have concentrated on providing information-theoretic privacy using database replications.

- *Computationally bounded PIR* (cPIR): If the PIR protocol involves a computationally bounded (or computationally intractable) database server entities then such scheme is considered as computationally bounded PIR (cPIR) in which the privacy is preserved based on the well-defined cryptographic intractability assumption(s). Most of the research work [6,11,20,23,25,33] and [7,9,16,22,26,29,31,35,36,38] on cPIR concentrated on using a single intractability assumption to preserve both user privacy and data privacy.

There are following major problems in the existing single database PBR schemes (including both itPBR and cPBR).

- Lack of sufficient itPIR approaches: More research focus was on the construction of an efficient cPBR instead of itPBR in a single database setting. This leads to the lack of information-theoretic privacy guarantee to the user in single database setting.
- Lack of independency between user and data privacy: Most of the existing cPBR schemes use a single intractability assumption (such as Quadratic residuosity, Phi-hiding, Lattices, Composite residuosity etc) to preserve both user privacy and data privacy. If the curious party breaks the underlying intractability assumption then both the privacy concerns are easily compromised without extra effort. For instance, the single database PIR protocol constructed by Kushilevitz and Ostrovsky [25] rely on the well-known intractability assumption called Quadratic Residuosity Assumption (QRA) to achieve both the *user privacy* (through the computationally intractable query inputs with quadratic residuosity properties) and the *data privacy* (through the quadratic residuosity ciphertexts). Note that compromising the QRA naturally reveals both privacy concerns (without extra effort). Therefore, there is a strong need of a generic scheme with efficient mapping from cPBR to itPBR in such a way that the underlying primitive of *user privacy* should also map from intractability assumption to information-theoretic privacy. Note that, Kushilevitz and Ostrovsky scheme does not support an efficient mapping cPBR to/from itPBR.
- Lack of generic framework that fulfills the above needs: Due to the lack of generic PBR framework (which can be used as a generic framework for several privacy critical applications such as PBR, oblivious transfer, asymmetric encryption etc), there is a strong need of a generic PBR scheme that can efficiently transform between several PBR extensions like information-theoretic PBR, compu-

tationally bounded PBR, oblivious transfer, asymmetric encryption etc.

With this thorough investigation, the natural question that arises is as follows.

Is it possible to construct a generic single database Private Block Retrieval framework with a reasonable performance that fulfills one or more privacy concerns (such as user privacy, data privacy, server privacy) of private block retrieval and oblivious transfer ?

Our single database private block retrieval solution: We have introduced a new bit connection and QRA based trapdoor functions for a single database PBR with the following results.

- New quadratic residuosity based single bit *injective* and *lossy* trapdoor functions.
- New bit connection methods (BCMs) called *rail-shape* and *signal-shape* to interconnect the proposed trapdoor functions with the aid of quadratic residuosity based injective trapdoor functions introduced by Freeman et al. [15].
- The appropriate combination of the proposed bit connection methods and trapdoor functions serve as a generic framework to map between several PBR extensions such as information-theoretic PBR, computationally bounded PBR, oblivious transfer, asymmetric encryption etc.
- New single database information-theoretic PBR (Sit PBR) scheme using the combination of proposed bit connection methods and trapdoor functions in which the communication cost of the proposed scheme is $\mathcal{O}(u(v-2) + 2u \log N)$ and its computation cost is $\mathcal{O}(u(2v-2))$ where $n = uv$ is the database size, u = rows, v = columns, and N is the RSA composite.
- New single database computationally bounded PBR (ScPBR) scheme in which the communication cost of the proposed scheme is $\mathcal{O}(u(v-2) + 2u \log N)$ and its computation cost is $\mathcal{O}(u(2v-2))$.

2 Related work

Many PIR schemes have also been extended to provide database security. Naor and Pinkas [34] have first proposed the transformation of cPIR to OT with a small computation overhead. Gertner et al. [18] have proposed the way of transforming cPIT to cOT with a small communication overhead. To achieve this, the system must add one more auxiliary database to store some random strings. Aiello et al. [1] have proposed the method of converting cPIR to cOT with an additional initialization phase which is not the component of classic OT method. Moreover the security has

been slightly relaxed from the standard notion. Chang [10] has proposed the first balanced computationally bounded OT scheme in single database setting. But, no method has been proposed to convert cOT to cPIR. Laur and Lipmaa [27] have proposed a disclose-if-equal (DIE) protocol which in turn supports cPIR to OT transformation. The proposed protocol involves compulsory client encryption operations, it is not best-suited for large database PIR operations. Kiayias et al. [24] have proposed rate-optimal cPIR-to-OT transformation. But, the transformed OT scheme is computationally secure on the server side. Therefore, authors left the information-theoretically server-private optimal-rate OT protocol as an interesting open problem.

But, there is no substantial efforts visible till date in the transformation of cPIR to itPIR. It is generally hard to achieve the sublinear communication in single database itPIR. But, it is essential to go with single database setting as multi-database setting provides weaker privacy guarantee. This basic reason motivated the design of the proposed transformation scheme.

3 Preliminaries and notations

Let $[1, u]$ denotes taking all values from 1 to u and $[u] \triangleq \{1, 2, \dots, u\}$ denotes taking any one value in the range from 1 to u . Let k denotes the security parameter, $N \leftarrow \{0, 1\}^k = PQ$ be the RSA composite modulus where $P \equiv 3 \pmod{4}$, $Q \equiv 3 \pmod{4}$, $\mathbb{Z}_N^{+1}$ denotes the set of all elements with *Jacobi Symbol* ($\mathcal{JS}$) 1. Let $\mathcal{Q}_R$ and $\overline{\mathcal{Q}}_R$ denote the quadratic residue and quadratic non-residue sets with $\mathcal{JS} = 1$ respectively. Let $\langle a, b \rangle$ be a set consists of two components in which $a \in \mathbb{Z}_N^{+1}$, and $b = \{i : i \in \{0, 1\}\}$.

Correctness: In any given instance, user must be able to retrieve the correct desired bit.

User privacy: In any given instance, user interest (may be in terms of database index) should not be revealed to the server.

Information theoretic privacy: Even after receiving the user query and having unbounded computation power, server should not gain any (even partial) information about the database index.

Quadratic residuosity predicate (QRP): $\forall x \in \mathbb{Z}_N^*$,

$$(QRP_{P,Q}(x) \text{ or } QRP(x)) = \begin{cases} 0 & \text{If } x \in \mathcal{Q}_R \\ 1 & \text{If } x \in \overline{\mathcal{Q}}_R \end{cases} \quad (1)$$

Quadratic residuosity based lossy trapdoor function of Freeman et al. [15] (LTDF): For all $\alpha \in \mathbb{Z}_N^*$, $s \in \overline{\mathcal{Q}}_R$ and $r \in \mathbb{Z}_N^{-1}$, the lossy trapdoor function $\mathcal{T} : \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is $\mathcal{T} = (\alpha^2 \cdot r^j \cdot s^h \equiv z \pmod{N})$ such that j is equal to 1 if $\mathcal{JS}(\alpha) = -1$ otherwise j is equal to 0. The value of h is equal to 1 if $\alpha > N/2$ otherwise h is equal to 0. The

Fig. 1 A single database private block retrieval framework with itPBR to/from cPBR transformations

respective inverse function is $\mathcal{T}^{-1} = (\sqrt{(z \cdot s^{-h}) \cdot r^{-j}} \equiv \alpha \pmod{N})$. We use the alternative square root syntax as $\mathcal{T}^{-1} = (\sqrt[j]{z} \equiv \alpha \pmod{N})$.

4 Combination of new bit connection methods and trapdoor functions

We have introduced a novel combinations of the quadratic residuosity based trapdoor functions in Sect. 4.1 and the database bit connection methods in Sect. 4.2 that can be used as a generic framework for itPBR to/from cPBR transformations as shown in Fig. 1. These combinations can assure many privacy concerns such as user privacy, data privacy and server privacy.

4.1 A new quadratic residuosity based trapdoor functions

It is a newly constructed 7-tuple $(\mathcal{I}, \mathcal{G}_0, \mathcal{G}_1, g, g^{-1}, f, f^{-1})$ consists of the following functions.

- **Sampling an input ($\mathcal{I}$):** The algorithm $\mathcal{I}$ receives the input 1^k and produces the large RSA composite $N = PQ$ where P and Q are large distinct primes with $P \equiv Q \equiv 3 \pmod{4}$ or $1 \pmod{4}$. Then chooses an “identically distributed” random $x \in \mathbb{Z}_N^{+1}$. The input domain of the random input x is $\mathbb{Z}_N^{+1}$.
- **Sampling a lossless injective function ($\mathcal{G}_0$):** On receiving the composite N , the algorithm $\mathcal{G}_0$ chooses a random $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{Z}_N^{+1}$ such that the quadratic residuosity predicate of $\mathcal{X}_1$ and $\mathcal{X}_2$ must be different (i.e., $QRP(\mathcal{X}_1) \neq QRP(\mathcal{X}_2)$). The function parameters are $\sigma = (N, \mathcal{X}_1, \mathcal{X}_2)$

and the trapdoor/private key is $\tau = (P, Q)$. Now it is clear that the injective function is defined over the domain $\mathbb{Z}_N^{+1}$.

- **Sampling a lossy trapdoor function** ($\mathcal{G}_1$): On receiving the composite N , the algorithm $\mathcal{G}_1$ chooses a random $\mathcal{X}_1, \mathcal{X}_2 \in \mathbb{Z}_N^{+1}$ such that the quadratic residuosity predicate of $\mathcal{X}_1$ and $\mathcal{X}_2$ must be equal (i.e., $\text{QRP}(\mathcal{X}_1) = \text{QRP}(\mathcal{X}_2)$).
- **Evaluation of trapdoor function of [15]** (g): The algorithm g receives the input x and produces “ h ” value of x (as described in quadratic residuosity based lossy trapdoor function [15]) as trapdoor bit as follows.

$$\begin{aligned} g(x) &= x^2 \pmod{N} \\ &= (x^2, h_x) \end{aligned} \quad (2)$$

- **Inversion of trapdoor function of [15]** (g^{-1}): Given the modular square x^2 and “ h ” value of x , the algorithm g^{-1} obtains the input x as follows.

$$g^{-1}(x^2, h_x) = \sqrt{jx=0, h_x}{x^2 \pmod{N}} = x \quad (3)$$

- **Evaluation of lossless injective function** (f): The algorithm f chooses a bit $b \in \{0, 1\}$. It then receives the function parameters, $g(x)$ and evaluates the following.

$$f_\sigma(g(x), b) = \begin{cases} g(x) \cdot \mathcal{X}_1 \pmod{N} & \text{If } b = 0 \\ g(x) \cdot \mathcal{X}_2 \pmod{N} & \text{If } b = 1 \end{cases} = y \quad (4)$$

- **Inversion of lossless injective function** (f^{-1}): Given the function parameters, trapdoor τ , trapdoor bit h and ciphertext y , the algorithm f^{-1} obtains both x and b as follows.

$$\begin{aligned} f_\tau^{-1}(y) &= \begin{cases} b = 0, g^{-1}(y \cdot \mathcal{X}_1^{-1}, h_x) & \text{If } \text{QRP}(\mathcal{X}_1) = \text{QRP}(y) \\ b = 1, g^{-1}(y \cdot \mathcal{X}_2^{-1}, h_x) & \text{If } \text{QRP}(\mathcal{X}_2) = \text{QRP}(y) \end{cases} \\ &= (x, b) \end{aligned} \quad (5)$$

where $\mathcal{X}_1 \cdot \mathcal{X}_1^{-1} \equiv 1 \pmod{N}$ and $\mathcal{X}_2 \cdot \mathcal{X}_2^{-1} \equiv 1 \pmod{N}$.

4.2 A new bit connection methods (BCMs)

We introduce new methods of interconnecting the database bits during PBR response creation on the server side as shown in Fig. 1. Based on the interconnectivity of the database bits, we classify the newly introduced bit connection methods as *rail-shape* and *signal-shape* as shown in Fig. 2.

Let the database be $\mathcal{DB} = \{b_1, b_2, \dots, b_v\}$. Consider the following ordered subsets of $\mathcal{DB}$

$$\begin{aligned} S_1 &= \{b_i : i = i + 2, i \in [1, v-1]\} \\ S_2 &= \{b_i : i = i + 2, i \in [2, v]\} \end{aligned} \quad (6)$$

Fig. 2 A new bit connection methods used to interconnect the proposed trapdoor functions

a Transition between PBR types in the existing PBR schemes (without BCMS)

b Transition between PBR types in the proposed PBR schemes (with BCMS)

Fig. 3 Possible transformations in the existing and the proposed PBR schemes

Note that if the absolute difference between any two database indices of the underlying set is 1 then such set is used for *rail-shape* connection and if the absolute difference between any two database indices of the underlying set is 2 then such set is used for *signal-shape* connection. Therefore, it is now intuitive that the set $\mathcal{DB}$ is used for rail-shape connection and S_1/S_2 are used for signal-shape connections.

Now, let's see the main advantage of using these BCMS in a single database PBR setting as follows.

- Most of the existing PBR schemes provide the whole database as input to their underlying trapdoor functions as shown in Fig. 3a. Consequently, this method of providing a database to the underlying trapdoor function in PBR results in the following types of PBR: either itPBR or cPBR. Also, there should always be a chance of transforming from each itPBR scheme to its cPBR version (i.e., $\text{Map}(\text{itPBR}) \rightarrow \text{cPBR}$). But, there is no chance of transforming from each cPBR scheme to its itPBR version (i.e., $\text{Map}(\text{cPBR}) \rightarrow \text{itPBR}$).
- Introducing the unique bit connection methods (other than using the whole plaintext) is helpful to achieve $\text{Map}(\text{itPBR}) \rightarrow \text{cPBR}$? Yes. It is possible to achieve both $\text{Map}(\text{itPBR}) \rightarrow \text{cPBR}$ and $\text{Map}(\text{cPBR}) \rightarrow \text{itPBR}$ using the combination of BCMS and newly constructed trapdoor

functions of Sect. 4.1 as shown in Fig. 3b. Therefore, the combination of BCMs and newly constructed trapdoor functions serve as a framework to construct either itPBR or cPBR and thereby achieving $\text{Map(itPBR)} \rightarrow \text{cPBR}$ and $\text{Map(cPBR)} \rightarrow \text{itPBR}$.

5 A new single database information-theoretic private block retrieval schemes (SitPBR)

In this section, we have introduced a new information-theoretic private block retrieval technique. At the abstract view, the proposed scheme is a 3-tuple (QG, RC, RR) involves two communicating parties: *user* and *server* in which *user* generates an information-theoretically private query from the input domain $\mathbb{Z}_N^{+1}$ using QG algorithm and sends this query to *server*. On the other hand, using query and the database $\mathcal{DB}$, *server* generates the response using RC algorithm and sends back to *user*. Finally, *user* retrieves the intended block privately using RR algorithm. The detailed description of the proposed scheme is given as follows.

Let $n = u \times v$ bit 2-dimensional matrix database with u rows and v columns be $\mathcal{DB} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_u\}$ where $\mathcal{D}_i = \{b_{i,1}, b_{i,2}, \dots, b_{i,v}\}$, $i \in [u]$. Each database block $\mathcal{D}_i = (S_2 \cup S_1)$ is further viewed as two subsets S_2 and S_1 where $S_2 = \{b_{i,2}, b_{i,4}, b_{i,6}, \dots, b_{i,v}\}$ and $S_1 = \{b_{i,1}, b_{i,3}, b_{i,5}, \dots, b_{i,v-1}\}$. The idea here is to use new bit connections using the subsets S_2, S_1 and apply the recursive execution of the proposed trapdoor function of Sect. 4.1. The detailed description of the proposed algorithms is given as follow.

- **Query generation (QG):** (*user* generates) Generate (public key, private key) pair from the query input domain $\mathbb{Z}_N^{+1}$ as follows. Generate the public key $\sigma = (N, \{\mathcal{K}_{z,1}, \mathcal{K}_{z,2} : z \in [1, u]\})$, and the private key $\tau = (P, Q)$ as described in the algorithm $\mathcal{G}_0$. Also, generate an “identically distributed” random $x \in \mathbb{Z}_N^{+1}$ as described in the algorithm $\mathcal{I}$. Then, generate an information-theoretically private query $\mathcal{Q} = (N, \{\mathcal{K}_{z,1}, \mathcal{K}_{z,2} : z \in [1, u]\}, x)$ where $\mathcal{Q}^z$ represents the z -th block query with public key components $(\mathcal{K}_{z,1}, \mathcal{K}_{z,2})$.
- **Response creation (RC):** (*server* generates) Using the information theoretic query $\mathcal{Q}$ and the database $\mathcal{DB}$, generate the response by executing the following. For all database block $\mathcal{D}_z$, $z \in [1, u]$, using respective public key components $\mathcal{K}_{z,1}, \mathcal{K}_{z,2}$, execute the following recursive function $f(g(\cdot), \cdot)$ as described in the algorithm f and obtain the intermediate ciphertext bits from each $g(\cdot)$ (as described in the algorithm g) and two final cipher-

texts as follows.

$$\begin{aligned} (y_{z,1}, \{s_{z,1}, s_{z,2}, \dots, s_{z, \frac{v}{2}-1}\}) &= f_{\sigma}(g(f_{\sigma}(\cdot, b_{z,i-2})), b_{z,i}) \\ \text{and} \\ (y_{z,2}, \{t_{z,1}, t_{z,2}, \dots, t_{z, \frac{v}{2}-1}\}) &= f_{\sigma}(g(f_{\sigma}(\cdot, b_{z,j-2})), b_{z,j}) \end{aligned} \quad (7)$$

where $i \in [v, 4]$, $j \in [v-1, 3]$ and each $f(g(\cdot), \cdot)$ is an injective function described in the algorithm f .

Finally, the database response would be $\mathcal{R} = \{\mathcal{R}_z = \{(y_{z,1}, \{s_{z,1}, s_{z,2}, \dots, s_{z, \frac{v}{2}-1}\}), (y_{z,2}, \{t_{z,1}, t_{z,2}, \dots, t_{z, \frac{v}{2}-1}\})\} : y \in \mathbb{Z}_N^{+1}, s, t \in \{0, 1\}\}$. The pictorial representation of the block response creation process is given in Fig. 4.

- **Response retrieval (RR):** (*user* generates) Using the response $\mathcal{R}$ and the trapdoor τ , retrieve the required block $w \in [u]$ (generally single block) as follows.

$$\begin{aligned} \{b_{w,2}, b_{w,4}, b_{w,6}, \dots, b_{w,v}\} &= f_{\tau}^{-1}(g^{-1}(f_{\tau}^{-1}(\cdot, s_{w,i}))) \\ \text{and} \\ \{b_{w,1}, b_{w,3}, b_{w,5}, \dots, b_{w,v-1}\} &= f_{\tau}^{-1}(g^{-1}(f_{\tau}^{-1}(\cdot, t_{w,i}))) \end{aligned} \quad (8)$$

where $i \in [\frac{v}{2}-1, 1]$ and each $f^{-1}(g^{-1}(\cdot), \cdot)$ is the inverse of the injective function described in Eq. (7). The pictorial representation of the response retrieval process is given in Fig. 4.

A toy example: Let $P = 23$, $Q = 17$ and $N = (P \cdot Q) = 391$, $\mathcal{K}_1 = 82$, $\mathcal{K}_2 = 10$, $x = 40$. Therefore the common query for both the blocks is $\mathcal{Q} = (391, \{82, 10\}, 40)$. Let a 2-dimensional two block matrix database be $\mathcal{DB} = \{\{0, 0, 1, 0, 1, 1, 0, 1\}, \{1, 0, 1, 1, 0, 0, 1, 1\}\}$ where $|\mathcal{DB}| = n = 16$, $u = 2$ and $v = 8$. Therefore, $\mathcal{D}_1 = \{0, 0, 1, 0, 1, 1, 0, 1\}$ and $\mathcal{D}_2 = \{1, 0, 1, 1, 0, 0, 1, 1\}$. Let us assume that the user is interested in the second block (i.e., $w = 2$). Note that the entire database response must be downloaded to retrieve any block. The complete illustrative example is given in Table 1.

6 A new single database computationally bounded private block retrieval schemes (ScPBR)

In this section, we have introduced a new computationally bounded block retrieval technique using computationally intractable queries. The response creation and the response retrieval algorithms are same as the SitPBR scheme. The detailed description of the query generation algorithm is given as follows.

Generate (public key, private key) pair from the query input domain $\mathbb{Z}_N^{+1}$ as follows. Let the user is interested in

Fig. 4 The block response creation (RC) and response retrieval (RR) algorithms of the proposed scheme

the database block $\mathcal{D}_w$, $w \in [u]$. Generate the first set of public key components $\{\mathcal{K}_{z,1}, \mathcal{K}_{z,2} : \forall z \in [u], z \neq w\}$, (from $\mathcal{G}_0$ algorithm) such that $\text{QRP}(\mathcal{K}_{z,1}) \neq \text{QRP}(\mathcal{K}_{z,2})$ and generate the second set of public key component pairs $(\mathcal{K}_{w,1}, \mathcal{K}_{w,2})$, $w \in [u]$ and $w \neq z$, (from $\mathcal{G}_1$ algorithm) such that $\text{QRP}(\mathcal{K}_{w,1}) = \text{QRP}(\mathcal{K}_{w,2})$. Note that these two sets of public key components are computationally intractable under quadratic residuosity assumption. The public key is $\sigma = (N, \{\mathcal{K}_{i,1}, \mathcal{K}_{i,2} : i \in [1, u]\})$, and private key is $\tau = (P, Q)$. Also, generate a random $x \in \mathbb{Z}_N^{+1}$. Then, generate a computationally intractable query $\mathcal{Q} = (N, \{\mathcal{K}_{i,1}, \mathcal{K}_{i,2} : i \in [1, u]\}, x)$ where $x \in \mathbb{Z}_N^{+1}$, $\mathcal{Q}^i$ represents the i -th block query with public key components $(\mathcal{K}_{i,1}, \mathcal{K}_{i,2})$.

7 Transformation (or mapping) of SitPBR to/from ScPBR without affecting the basic setup

Most of the existing single database PBR schemes are concentrated on constructing single type of PBR either itPBR or cPBR. But, what if somebody wants to covert from one type to another without changing the basic setup? Essentially, there should be a framework of techniques that provides both types and the transformation mechanism between them.

In order to provide the above mentioned generic framework, we have proposed single database itPBR schemes in Sect. 5, single database cPBR in Sect. 6. Now, we describe the transformation of one type to another without changing the basic setup as follows.

The transformation of the proposed SitPBR to/from ScPBR depends upon the appropriate quadratic residuosity properties of the public key components. If so, how to choose the appropriate property public key components in the proposed PBR? Just look into the following descriptions to find the answer to this.

- **Sampling function parameters for SitPBR ($\mathcal{L}_0$)** : The algorithm $\mathcal{L}_0$ chooses the *identically distributed* public key components $\mathcal{K}_1, \mathcal{K}_2$ from $\mathbb{Z}_N^{+1}$ such that $\text{QRP}(\mathcal{K}_1) \neq \text{QRP}(\mathcal{K}_2)$ (as described in the algorithm $\mathcal{G}_0$) during QG algorithm execution without altering the remaining algorithms.
- **Sampling function parameters for ScPBR ($\mathcal{L}_1$)** : The algorithm $\mathcal{L}_1$ chooses both kinds of public key components from $\mathcal{G}_0$ and $\mathcal{G}_1$ algorithms during QG algorithm execution such that both kinds of components are computationally indistinguishable. Note that choosing these appropriate property public key components neither affects the remaining PBR algorithms nor effect the basic PBR setup.

Table 1 An illustrative example of the proposed scheme with two database blocks

Step	Response Creation (RC)			
	$\mathcal{D}_1$	$\mathcal{D}_2$	$\mathcal{D}_1$	$\mathcal{D}_2$
	$\mathcal{S}_2 = \{0, 0, 1, 1\}$	$\mathcal{S}_1 = \{0, 1, 1, 0\}$	$\mathcal{S}_2 = \{0, 1, 0, 1\}$	$\mathcal{S}_1 = \{1, 1, 0, 1\}$
1	$f_1(40, 0) = 152$	$f_1(40, 0) = 152$	$f_1(40, 0) = 152$	$f_1(40, 1) = 9$
2	$g_2(152) = \langle 35, 0 \rangle$	$g_2(152) = \langle 35, 0 \rangle$	$g_2(152) = \langle 35, 0 \rangle$	$g_2(9) = \langle 81, 0 \rangle$
3	$f_2(35, 0) = 133$	$f_2(35, 1) = 350$	$f_2(35, 1) = 350$	$f_2(81, 1) = 28$
4	$g_3(133) = \langle 94, 0 \rangle$	$g_3(350) = \langle 117, 1 \rangle$	$g_3(350) = \langle 117, 1 \rangle$	$g_3(28) = \langle 2, 0 \rangle$
5	$f_3(94, 1) = 158$	$f_3(117, 1) = 388$	$f_3(117, 0) = 210$	$f_3(2, 0) = 164$
6	$g_4(158) = \langle 331, 0 \rangle$	$g_4(388) = \langle 9, 1 \rangle$	$g_4(210) = \langle 308, 1 \rangle$	$g_4(164) = \langle 308, 0 \rangle$
7	$f_4(331, 1) = 182$ $(182, \{0, 0, 0\})$ $\mathcal{R}_1 = \{(182, \{0, 0, 0\}), (347, \{0, 1, 1\})\}$	$f_4(9, 0) = 347$ $(347, \{0, 1, 1\})$	$f_4(308, 1) = 343$ $(343, \{0, 1, 1\})$ $\mathcal{R}_2 = \{(343, \{0, 1, 1\}), (343, \{0, 0, 0\})\}$	$f_4(308, 1) = 343$ $(343, \{0, 0, 0\})$
Step	Response Retrieval (RR)			
	$\mathcal{R}_1 = \{(182, \{0, 0, 0\}), (347, \{0, 1, 1\})\}$	$\mathcal{R}_2 = \{(343, \{0, 1, 1\}), (343, \{0, 0, 0\})\}$	$\mathcal{R}_1$	$\mathcal{R}_2$
1	$f_4^{-1}(182) = \langle 331, 1 \rangle$	$f_4^{-1}(347) = \langle 9, 0 \rangle$	$f_4^{-1}(343) = \langle 308, 1 \rangle$	$f_4^{-1}(343) = \langle 308, 1 \rangle$
2	$g_3^{-1}(331, 0) = 158$	$g_3^{-1}(9, 1) = 388$	$g_3^{-1}(308, 1) = 210$	$g_3^{-1}(308, 0) = 164$
3	$f_3^{-1}(158) = \langle 94, 1 \rangle$	$f_3^{-1}(388) = \langle 117, 1 \rangle$	$f_3^{-1}(210) = \langle 117, 0 \rangle$	$f_3^{-1}(164) = \langle 2, 0 \rangle$
4	$g_2^{-1}(94, 0) = 133$	$g_2^{-1}(117, 1) = 350$	$g_2^{-1}(117, 1) = 350$	$g_2^{-1}(2, 0) = 28$
5	$f_2^{-1}(133) = \langle 35, 0 \rangle$	$f_2^{-1}(350) = \langle 35, 1 \rangle$	$f_2^{-1}(350) = \langle 35, 1 \rangle$	$f_2^{-1}(28) = \langle 81, 1 \rangle$
6	$g_1^{-1}(35, 0) = 152$	$g_1^{-1}(35, 0) = 152$	$g_1^{-1}(35, 0) = 152$	$g_1^{-1}(81, 0) = 9$
7	$f_1^{-1}(152) = \langle 40, 0 \rangle$ $\mathcal{S}_2 = \{0, 1, 0, 1\}$ $\mathcal{D}_1$	$f_1^{-1}(152) = \langle 40, 0 \rangle$ $\mathcal{S}_1 = \{1, 1, 0, 1\}$	$f_1^{-1}(152) = \langle 40, 0 \rangle$ $\mathcal{S}_2 = \{0, 1, 0, 1\}$ $\mathcal{D}_2$	$f_1^{-1}(9) = \langle 40, 0 \rangle$ $\mathcal{S}_1 = \{1, 1, 0, 1\}$

- **Sampling function parameters for Map(Sit PBR)**
→ **ScPBR** ($\mathcal{M}_0$): In order to map from proposed SitPBR to ScPBR, just choose the appropriate public key components from $\mathcal{L}_1$ during QG algorithm execution and continue to execute the remaining PBR algorithms. Note that this mapping process is computationally indistinguishable.
- **Sampling function parameters for Map(ScP BR)**
→ **SitPBR** ($\mathcal{M}_1$): In order to map from proposed ScPBR to SitPBR, just choose the appropriate public key components from $\mathcal{L}_0$ during QG algorithm execution and continue to execute the remaining PBR algorithms (as usual). Note that this mapping process is also computationally indistinguishable.

8 Performance evaluation

Privacy: The proposed scheme of Sect. 5 always preserves the *user privacy* against the *curious-server* through the generation of information-theoretically private queries. If $\mathcal{Q}_1 = (N, \mathcal{K}_1, \mathcal{K}_2, x_1)$, $\mathcal{Q}_2 = (N, \mathcal{K}_3, \mathcal{K}_4, x_2)$ are any two randomly generated queries in QG algorithm then the selection of public key components from the identically distributed

domain for all database blocks always guarantees perfect user privacy i.e., the query components are randomly chosen from an identically distributed domain in such a way that the *mutual information* between any two queries is always zero and assures *perfect privacy* to the user.

The proposed scheme of Sect. 6 always preserves the *user privacy* against the *curious-server* through the generation of computationally bounded queries. If $\mathcal{Q}_1 = (N, \mathcal{K}_1, \mathcal{K}_2, x_1)$, $\mathcal{Q}_2 = (N, \mathcal{K}_3, \mathcal{K}_4, x_2)$ are any two randomly generated queries in QG algorithm then the computationally indistinguishable selection of public key components for all database blocks always guarantees computationally bounded user privacy. In other words, the quadratic residuosity properties of public-key components of $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are computationally hidden from the curious server.

Both the proposed schemes of Sects. 5 and 6 use quadratic residuosity assumption to preserve *data privacy* against intermediate adversary.

Communication and Computation: In the proposed schemes of Sects. 5 and 6, user sends $\mathcal{O}((2u + 2) \cdot \log N)$ query bits to the server. The server sends $\mathcal{O}(u(v - 2) + 2u \log N)$ response bits to the user where u is the row size of the database, v is the column size of the database, N is the composite modulus. Both the proposed schemes are single

Fig. 5 Proposed privacy preserving big data access control models

round PBR protocols use only one request-response cycle where user requests for a database block and server responds through the response.

The execution of the RC algorithms of the Sect. 5 and Sect. 6 involve uv number of lossless trapdoor functions $f(\cdot)$ and $u(v-2)$ number of lossy trapdoor functions $g(\cdot)$. Each trapdoor function (either lossy or lossless) involves a single modular multiplication, the RC algorithm involves a total of $u(2v-2)$ number of modular multiplications. On the other hand, the RR algorithms of Sects. 5 and 6 involve only $(2v-2)$ number of modular multiplications plus $(v-2)$ number of quadratic square roots to retrieve the required block.

9 Privacy preserving big data access control

We will extend our work to introduce a novel privacy preserving access control model in Big Data information processing environment. The core idea is to store only the CCA secure ciphertext components of the proposed PBR schemes of Sect. 5 or Sect. 6 on the Big Data and download the stored information using one of the proposed PBR techniques in 2-party and 3-party scenarios as shown in Fig. 5a, b. This idea covers many privacy critical applications such as Healthcare, Patent and Stock search, Email, Social media, Private chat which cannot be handled by traditional Big Data information processing model alone.

In 2-party scenario, the proposed model consists of two communicating parties: *Alice* and *Cloud* in which *Alice* encrypts his/her data $\mathcal{DB}$ using his/her own public key σ using RC algorithm and stores one of the ciphertext components $C_1 = \{(s_{z,1}, s_{z,2}, \dots, s_{z, \frac{v}{2}-1}), (t_{z,1}, t_{z,2}, \dots, t_{z, \frac{v}{2}-1})\}$, $\forall z \in [1, u]$ on *Cloud* (which maintains Big Data storage and processing) and keeps other ciphertext components

$C_2 = (y_{z,1}, y_{z,2}), \forall z \in [1, u]$ with him/her. Whenever required, *Alice* directly downloads partial ciphertext component C_1 from *Cloud* or downloads using the proposed schemes of Sects. 5, 6 and decrypts his/her data $\mathcal{DB}$ using his/her own private key τ using RR algorithm.

In 3-party scenario, the proposed model consists of three communicating parties: *Alice*, *Bob* and *Cloud* in which *Alice* encrypts his/her data using *Bob's* public key σ using RC algorithm and stores one of the ciphertext components $C_1 = \{(s_{z,1}, s_{z,2}, \dots, s_{z, \frac{v}{2}-1}), (t_{z,1}, t_{z,2}, \dots, t_{z, \frac{v}{2}-1})\}$, $\forall z \in [1, u]$ on *Cloud* (which maintains Big Data storage and processing) and sends other ciphertext component $C_2 = (y_{z,1}, y_{z,2}), \forall z \in [1, u]$ to *Bob*. Whenever required, *Bob* downloads a part of ciphertext components C_2 from *Alice* and downloads other ciphertext component C_1 from *Cloud* and decrypts *Alice's* data using his/her private key τ using RR algorithm.

10 Conclusion and future work

We have presented a new combination of trapdoor functions and bit connection methods to achieve a novel mapping single database information-theoretic and computationally bounded private block retrieval schemes and their transformations. Although, the proposed schemes show reasonable performance with the current state-of-art work, focusing on other dimensions such as scalable and fault-tolerant multi-server PBR scheme for practical privacy-preserving BigData access control applications is the future direction.

Funding Open access funding provided by Manipal Academy of Higher Education, Manipal.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as

long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Aiello B, Ishai Y, Reingold O (2001) Priced oblivious transfer: How to sell digital goods. In: Pfitzmann B (ed) *Advances in cryptography—EUROCRYPT 2001*, Lecture Notes in Computer Science, vol 2045. Springer, New York, pp 119–135
2. Banawan K, Ulukus S (2018) Multi-message private information retrieval: capacity results and near-optimal schemes. *IEEE Trans Inf Theory* 64(10):6842–6862. <https://doi.org/10.1109/TIT.2018.2828310>
3. Beimel A, Ishai Y, Kushilevitz E (2005) General constructions for information-theoretic private information retrieval. *J Comput Syst Sci* 71(2):213–247
4. Beimel A, Stahl Y (2007) Robust information-theoretic private information retrieval. *J Cryptol* 20(3):295–321
5. Benny C, Niv G, Moni N (1998) Private information retrieval by keywords. *Cryptology ePrint Archive*, Report 1998/003. <http://eprint.iacr.org/1998/003>
6. Cachin C, Micali S, Stadler M (1999) Computationally private information retrieval with polylogarithmic communication. In: *Proceedings of 17th theory and application of cryptographic techniques, EUROCRYPT'99*. Springer, New York, pp 402–414
7. Cachin C, Micali S, Stadler M (1999) Computationally private information retrieval with polylogarithmic communication. Springer, Berlin, pp 402–414
8. Chakrabarti A, Shubina A (2007) Nearly private information retrieval. In: *MFCS 2007*, pp 383–393
9. Chang Y.C (2004) Single database private information retrieval with logarithmic communication. Springer, New York, pp 50–61
10. Chang YC (2004) Single database private information retrieval with logarithmic communication. *Cryptology ePrint Archive*, Report 2004/036. <http://eprint.iacr.org/2004/036>
11. Chor B, Gilboa N (1997) Computationally private information retrieval (extended abstract). In: *Proceedings of 29th STOC, STOC '97*. ACM, pp 304–313
12. Chor B, Goldreich O, Kushilevitz E, Sudan M (1995) Private information retrieval. In: *Proceedings of the 36th FOCS, FOCS '95*. IEEE Computer Society, pp 41–50
13. Chor B, Kushilevitz E, Goldreich O, Sudan M (1998) Private information retrieval. *J ACM* 45(6):965–981
14. Demmler D, Herzberg A, Schneider T (2014) Raid-PIR: practical multi-server PIR. In: *Proceedings of the 6th edition of the ACM workshop on cloud computing security, CCSW '14*, pp 45–56
15. Freeman DM, Goldreich O, Kiltz E, Rosen A, Segev G (2009) More constructions of lossy and correlation-secure trapdoor functions. *Cryptology ePrint Archive*, Report 2009/590. <http://eprint.iacr.org/2009/590>
16. Gentry C, Ramzan Z (2005) Single-database private information retrieval with constant communication rate. In: *Proceedings of 32nd ICALP, ICALP'05*, pp 803–815. Springer, New York
17. Gertner Y, Ishai Y, Kushilevitz E, Malkin T (1998) Protecting data privacy in private information retrieval schemes. In: *STOC '98*. ACM, pp 151–160
18. Gertner Y, Ishai Y, Kushilevitz E, Malkin T (1998) Protecting data privacy in private information retrieval schemes. In: *Proceedings of 13th STOC, STOC '98*. ACM, pp 151–160
19. Goldberg I (2007) Improving the robustness of private information retrieval. In: *IEEE symposium on security and privacy*, pp 131–148
20. Groth J, Kiayias A, Lipmaa H (2010) Multi-query computationally-private information retrieval with constant communication rate. In: *Proceedings of 13th PKC, PKC'10*. Springer, New York, pp 107–123
21. Henry R (2016) Polynomial batch codes for efficient IT-PIR. *Cryptology ePrint Archive*, Report 2016/598. <https://eprint.iacr.org/2016/598>
22. Henry R, Huang Y, Goldberg I (2013) One (block) size fits all: PIR and SPIR with variable-length records via multi-block queries
23. Ishai Y, Kushilevitz E, Ostrovsky R, Sahai A (2006) Cryptography from anonymity. In: *Proceedings of 47th FOCS, FOCS '06*. IEEE Computer Society, pp 239–248
24. Kiayias A, Leonardos N, Lipmaa H, Pavlyk K, Tang Q (2015) Optimal rate private information retrieval from homomorphic encryption. *Proc Priv Enhanc Technol* 2015(2):222–243. <https://doi.org/10.1515/popets-2015-0016>
25. Kushilevitz E, Ostrovsky R (1997) Replication is not needed: single database, computationally-private information retrieval. In: *Proceedings of 38th FOCS, FOCS '97*. IEEE Computer Society, p 364
26. Kushilevitz E, Ostrovsky R (2000) One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In: *Proceedings of 19th theory and application of cryptographic techniques, EUROCRYPT'00*. Springer, New York, pp 104–121
27. Laur S, Lipmaa H (2005) A new protocol for conditional disclosure of secrets and its applications. *ACNS 2007*, h.lipmaa@cs.ucl.ac.uk 13592, received 20 Oct 2005, last revised 20 Mar 2007
28. Li Y, Chang Y, Cheng M, Feng T (2021) Multi-value private information retrieval with colluding databases via trace functions. *Inf Sci* 543:426–436
29. Li Z, Ma C, Wang D, Du G (2017) Toward single-server private information retrieval protocol via learning with errors. *J Inf Secur Appl* 34:280–284
30. Lipmaa H (2005) An oblivious transfer protocol with log-squared communication. In: *Proceedings of 8th ISC, ISC'05*. Springer, New York, pp 314–328
31. Lipmaa H (2010) First CPPIR protocol with data-dependent computation. In: *Proceedings of 12th information security and cryptography, ICISC'09*. Springer, New York, pp 193–210
32. Liu T, Vaikuntanathan V (2016) On basing private information retrieval on np-hardness. In: *TCC 2016-A*, pp 372–386
33. Melchor CA, Gaborit P (2007) A lattice-based computationally-efficient private information retrieval protocol
34. Naor M, Pinkas B (1999) Oblivious transfer and polynomial evaluation. In: *Proceedings of the thirty-first annual ACM symposium on theory of computing, STOC '99*. Association for Computing Machinery, pp 245–254. <https://doi.org/10.1145/301250.301312>
35. Rebollo-Monedero D, Forne J, Solanas A, Martínez-Balleste A (2010) Private location-based information retrieval through user collaboration. *Comput Commun* 33(6):762–774
36. Trostle J, Parrish A (2011) Efficient computationally private information retrieval from anonymity or trapdoor groups. In: *Proceedings of 13th ISC, ISC'10*. Springer, New York, pp 114–128
37. Wang L, Kuppasamy TK, Liu Y, Capps J (2015) A fast multi-server, multi-block private information retrieval protocol. In: *2015 IEEE global communications conference (GLOBECOM)*, pp 1–6. <https://doi.org/10.1109/GLOCOM.2015.7417246>
38. Zhao L, Wang X, Huang X (2021) Verifiable single-server private information retrieval from LWE with binary errors. *Inf Sci* 546:897–923