A Novel Tamper Evident Single Database Information-Theoretic Private Information Retrieval for User Privacy Applications

Radhakrishna Bhat(✉) and N R Sunitha

Department of Computer Science and Engineering,
Siddaganga Institute of Technology,
Affiliated to Visvesvaraya Technological University,
B H Road, Tumakuru, Karnataka, India 572103
{rsb567@gmail.com, nrsunithasit@gmail.com}

Abstract. Providing *perfect privacy* to the user against analytics enabled *trusted-but-curious* type of database server during private information retrieval has gained a major attention. The major problem with the existing *user privacy* preserving information retrieval methods is that either server has adopted its own privacy preserving policy (i.e., user privacy is guaranteed through the server privacy policy) or user has conveyed to use intractability assumption based user privacy preserving techniques. Due to this, user privacy is not completely assured till date. We have successfully constructed a perfect user privacy preserving information retrieval scheme in a single database setting called sitPIR using the concept of Private Information Retrieval (PIR). In the proposed scheme, the identically distributed $\mathcal{O}(5 \log N)$ bits query exhibit *perfect privacy* where N is the RSA composite. Note that the proposed scheme preserves *user privacy* (i.e., user interest) using information-theoretic query against *curious server* and preserves *data privacy* through $\mathcal{O}(o(n)+$ 2log $N)$ response bits against computationally bounded *intermediate adversary* using Quadratic Residuosity Assumption (QRA) where n is the database size. We have also extended the proposed scheme to a tamper evident single database information-theoretic Private Block Retrieval (PBR) scheme called sitPBR.

Keywords: Private information retrieval · Information-theoretic· Perfect privacy · Quadratic residuosity · Private Block Retrieval · Tamper evident retrieval

1 Introduction

Consider a scenario where the user wants to retrieve a record of information (may be a single bit or block) from a single database server privately without revealing any information about the record retrieved.

Private Information Retrieval (PIR)[10] is one of the user privacy preserving techniques and involves two participating entities: *User* and *Server* in which the

User wants to retrieve or read a bit from the *Server* without revealing his interest. Hence, PIR is the way of retrieving the required information through the database reference (the reference maybe the index or the address of the information stored on the server) from the *Server* by hiding the reference. The primary concern in any PIR protocol is to hide the reference from the server along with reading the required bit from the database server. This concept was introduced in a replicated database setting and coined as private information retrieval by Chor et.al [10,12]. Private Block Retrieval (PBR) is the realistic version of PIR in which the user retrieves a block from the set of blocks maintained by the server.

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 user privacy is preserved based on the well-defined cryptographic intractability assumption(s) based queries. If the PIR protocol involves non-colluding replicated database server entities then such scheme is considered as information-theoretic PIR (itPIR) in which the user privacy is preserved based on the information-theoretically private queries.

Most of the existing computationally bounded PIR schemes including ϕ-hiding assumption based scheme proposed by Cachin et.al [6], Pailliers cryptosystem based scheme proposed by Chang [9], one-way function based scheme proposed by Chor and Gilboa [11], decision subgroup problem called ϕ-hiding assumption based scheme proposed by Gentry and Ramzan [14], the multi query scheme introduced by Groth et.al [17], anonymity technique based scheme proposed by Ishai et.al [18], quadratic residuosity assumption based scheme introduced by Kushilevitz and Ostrovsky [19], one-way trapdoor permutation based scheme proposed by Kushilevitz and Ostrovsky [20], composite residuosity assumption based scheme presented by Lipmaa [22], Coding theory and Lattice assumption based PIR scheme is also presented by Aguilar-Melchor and Gaborit [24], trapdoor group based scheme presented by Jonathan and Andy [25], Lattice based scheme presented by Aguilar-Melchor et.al [1] and preprocessing based scheme presented by Canetti et. al [7] are all involved a single intractability assumption to preserve both *user privacy* and *data privacy*.

Several information-theoretic schemes [3,23,8,15,16,4,2] and PBR extension schemes [12,5,21,14,22] proposed in PIR environment are also suffering from the imbalance between the communication cost and resource utilization, involvement of additional techniques (like pre-processing, caching etc) and multiple rounds, inability to achieve the non-trivial communication cost in a single database environment.

PROBLEMS WITH EXISTING PIR SCHEMES: There are two major problems in the existing single database PIR schemes as described below.

- Much attention is given on computationally protecting the privacy of the user using a well-defined intractability assumption that the database server is computationally bounded (i.e., database server has limited computation capability). All such cryptographic intractability assumption based privacy

preserving techniques fail to provide *user privacy* if the database server attains high computational power.
– All the single database PIR schemes rely on a single intractability assumption (like Phi-hiding, Lattice, QRA, Composite Residuosity and n-th residuosity, one-way Functions etc) to preserve both *user privacy* (assuming that the curious server is computationally bounded) and *data privacy* (assuming that the intermediate adversary is computationally bounded). Intuitively, if the adversary is able to reveal any one of them, he/she will get the other without any extra effort.

In order to provide perfect user privacy preserving single database PIR solution, the best way is that all the successively generating queries must be *identically distributed*. Therefore, all the existing single database PIR schemes have clearly failed to generate the identically distributed queries due to the existence of underlying intractability assumption.

PERFECT USER PRIVACY PRESERVING PIR SOLUTION: Conventionally, we use the term "server" for database server, we use the term "privacy" for user privacy, we use the term "perfect privacy" for perfect user privacy unless and until externally stated. We also use the terms "perfect privacy" and "information-theoretic privacy" interchangeably.

We have constructed an information-theoretic query which takes identically distributed random input from the quadratic residuosity set $\mathbb{Z}_N^{+1}$ for preserving *user privacy* (Note that in the QRA based Kushilevitz and Ostrovsky [19] scheme, the query input for the interested bit is always drawn from quadratic non-residue. Therefore, all such randomly generated queries are not identically distributed.) and new QRA based recursive 2-bit encryptions called *pair-link encryption* (PLF) (which encrypts two bits and decrypts one bit at a time) for preserving the *data privacy*. Also, we have introduced new methods of selection and encryption of database bits called *criss-cross encryption* (CCE) and *snake-walk encryption* (SWE) using the pair-link encryption as the basic building block during PIR invocation.

With the aid of the proposed pair-link encryption and CCE/SWE methods, we have successfully constructed the perfect privacy preserving single database PIR scheme with the following results.

– All the random queries generated in the proposed scheme are identically distributed over $\mathbb{Z}_N^{+1}$ and hence, exhibit *information-theoretic* privacy over the trusted-but-curious server (i.e., *user privacy* is always guaranteed and independent of the security parameter). That is, an individual query or a randomly selected pair of queries gives no information (not even a partial information) about the user interest.
– The proposed scheme uses quadratic residuosity as the underlying *data privacy* primitive to preserve the communicating data over the intermediate adversary.
– The overall communication cost is $(o(n)+2 \log N)$ where n is the database size, N is the RSA composite modulus. The communication cost can reach

non-triviality (i.e., less than the database size) for all $c_0 > c$ and n=2^{c_0} where c is an integer constant.
- Inbuilt tamper evidence for the communicating data (from server to user) when the PBR version of the proposed scheme is used.
- The proposed scheme can easily be extended to oblivious transfer and computationally bounded PIR schemes.

ORGANIZATION: All the necessary preliminaries and notations are described in Section 2. The proposed information-theoretic PIR scheme along with required building blocks, security proofs, the performance details and the PBR version are described in Section 3. Finally, the open problems are described along with the conclusion in Section 4.

2 Preliminaries and Notations

2.1 Notations

Let $[u] \triangleq \{1, 2, \cdots, u\}$, Let k denote the security parameter, $N \xleftarrow{R} \{0,1\}^k$=$pq$ be the RSA composite modulus where $p \equiv 3 \pmod 4$,$q \equiv 3 \pmod 4$, $\mathbb{Z}_N^{+1}$ denote the set of all elements with *Jacobi Symbol* ($\mathcal{JS}$) 1. Let Q_R and $\overline{Q}_R$ denote the quadratic residue and quadratic non-residue sets with $\mathcal{JS}$=1 respectively. Let $< a, b >$ be two components set where $a \in \mathbb{Z}_N^{+1}$ and b=$\{i : i \in \{0,1\}\}$.

2.2 Preliminaries

Quadratic residuosity: For any element $a \in \mathbb{Z}_N^*$ if there exists an element b^2 congruent to a modulo N then a is called the quadratic residue otherwise quadratic non-residue modulo N. Intuitively, $\mathcal{JS}$ is equal to 1 for all elements that belongs to $\mathbb{Z}_N^{+1}$ and $\mathcal{JS}$ is equal to -1 for all elements that belongs to $\mathbb{Z}_N^{-1}$ where $\mathcal{JS}(\cdot)$ is the *Jacobi Symbol* modulo N.
Quadratic Residuosity Predicate (QRP): $\forall a \in \mathbb{Z}_N^*$, QRP is a function to return a value (0 or 1) to indicate whether a is Q_R if $QRP_{p,q}(a)$=0 or $\overline{Q}_R$ if $QRP_{p,q}(a)$=1.
Quadratic Residuosity Assumption (QRA): For all $N \in \{0,1\}^k$, for all $\mathcal{R} \in \mathbb{Z}_N^{+1}$, for all probabilistic polynomial time intermediate adversary Ad, PROB[$Ad(N,\mathcal{R})$= $QRP_N(\mathcal{R})$]< p^{QR} where p^{QR}=(1/2)+($1/k^c$) and c is a constant.
Quadratic residuosity based lossy trapdoor function of Freeman et.al [13] (LTDF): For all $\alpha \in \mathbb{Z}_N^*$, $s \in \overline{Q}_R$ and $r \in \mathbb{Z}_N^{-1}$, the lossy trapdoor function $\mathcal{T}$: $\mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$ is $\mathcal{T}$=($\alpha^2 \cdot r^{jx} \cdot s^{hx} \equiv z \pmod N$) such that jx is equal to 1 if $\mathcal{JS}(\alpha)$=-1 otherwise jx is equal to 0. The value of hx is equal to 1 if $\alpha > N/2$ otherwise hx is equal to 0. The respective inverse function is $\mathcal{T}^{-1}$=($\sqrt{(z \cdot s^{-hx}) \cdot r^{-jx}} \equiv \alpha \pmod N$). We use the alternative square root syntax as $\mathcal{T}^{-1}$=($\sqrt[jx,hx]{z} \equiv \alpha \pmod N$).

3 A Single Database Information-Theoretic Private Information Retrieval (sitPIR)

Definition 1 *A Single database information-theoretic PIR (sitPIR):* It is a 4-tuple (KG,QF,RC, IE) protocol that involves two communicating parties: user $\mathcal{U}_{pir}$ and server $\mathcal{S}_{pir}$ in which $\mathcal{S}_{pir}$ maintains n bit single dimensional matrix database $\mathcal{DB}$= $\{b_1,b_2,\cdots,b_n\}$. User $\mathcal{U}_{pir}$ requests the interested bit b_i, $i \in [n]$, privately from $\mathcal{S}_{pir}$ by generating information-theoretic query $\mathcal{Q}$ such that $\forall b_i$, $b_j \in \mathcal{DB}$, $i, j \in [n]$, any two random generated queries $\mathcal{Q}_i$ and $\mathcal{Q}_j$ exhibit same level of information-theoretic privacy equivalent to *perfect privacy* as described by Chor et.al [12] and the server $\mathcal{S}_{pir}$ in-turn generates the response $\mathcal{R}$ with the communication $\mathcal{O}(o(n)+2 \log N)$. The setting consists of the following polynomial time algorithms.

1. *Key Generation* (KG): $\mathcal{U}_{pir}$ calculates RSA modulus $N \xleftarrow{R} \{0,1\}^k$. $\mathcal{U}_{pir}$ then generates (public,private) key pair $(pk, sk) \xleftarrow{R} KG(1^k)$ where pk=(N, ($\mathcal{PK}_1$, $\mathcal{PK}_2$) $\in \mathbb{Z}_N^{+1}$) and sk=(p, q).
2. *Query Formulation* (QF): $\mathcal{U}_{pir}$ generates (pk, sk) from the key generation algorithm KG. $\mathcal{U}_{pir}$ then generates the perfect privacy preserving query as $\{(\mathcal{Q} = (\alpha, pk), sk) \leftarrow QF(1^k) : \alpha \xleftarrow{R} \mathbb{Z}_N^{+1}\}$ and keeps sk secret. Importantly, the random generation of the index (or reference) independent input α from either Q_R or $\overline{Q}_R$ always exhibits *perfect privacy* (Note: All the query input α randomly selected from $\mathbb{Z}_N^{+1}$ are always domain independent. Also, note that the "domain" here is the quadratic residuosity sets like Q_R and $\overline{Q}_R$).
3. *Response Creation* (RC): $\mathcal{S}_{pir}$ generates the response $\mathcal{R}$ using the query $\mathcal{Q}$ and the database $\mathcal{DB}$ as $\mathcal{R} \leftarrow RC(\mathcal{Q}, \mathcal{DB}, n, 1^k)$.
4. *Interest Extraction* (IE): Using the response $\mathcal{R}$ and the secret sk, $\mathcal{U}_{pir}$ extracts the required bit b_i, $i \in [n]$, as $b_i \leftarrow IE(\mathcal{R}, sk, n, 1^k)$.

Definition 2 *(Single database information-theoretic PIR (sitPIR))***:** Let $\mathcal{DB}$= $\{b_1,b_2,\cdots,b_n\}$ be n bit server database. Let query formulation QF, response creation RC of Definition 1 be the Probabilistic Polynomial Time (PPT) algorithms and interest extraction IE be the deterministic polynomial time algorithm. We say that 4-tuple (KG,QF,RC,IE) protocol of Definition 1 is a single database perfect privacy PIR scheme or single database itPIR scheme if the following conditions hold.

1. *Perfect user privacy:* For any two identically distributed random queries $\mathcal{Q}_i$ and $\mathcal{Q}_j$, $i, j \in [n]$,
   PROB[$(\mathcal{Q}_i,sk) \xleftarrow{R} QF(1^k)$: $Adv(n,\mathcal{Q}_i,1^k)$=1]= PROB[$(\mathcal{Q}_j,sk) \xleftarrow{R} QF(1^k)$: $Adv(n,\mathcal{Q}_j,1^k)$=1]
2. *Correctness:* $\forall z \in [n]$, PROB[$IE\big((\mathcal{R},sk,n,1^k)$:$\mathcal{R} \xleftarrow{R} RC(\mathcal{Q}_z,\mathcal{DB},n,1^k)$,$(\mathcal{Q}_z$, $sk) \xleftarrow{R} QF(1^k)\big)$=$b_z$]=1
3. *Data privacy:* For any two randomly selected queries $\mathcal{Q}_1$ and $\mathcal{Q}_2$, for all security parameter k and for all ciphertexts $\mathcal{R}_1$, $\mathcal{R}_2$,

$|\text{PROB}[(\mathcal{Q}_1,sk)\xleftarrow{R} QF(1^k),\mathcal{R}_1 \xleftarrow{R} RC(\mathcal{Q}_1,\mathcal{DB},n,1^k) : Ad(n,\ \mathcal{R}_1,\ 1^k)=1]-$
$\text{PROB}[(\mathcal{Q}_2,sk)\xleftarrow{R} QF(1^k),\mathcal{R}_2 \xleftarrow{R} RC(\mathcal{Q}_2,\ \mathcal{DB},\ n,\ 1^k) : Ad(n,\ \mathcal{R}_2,\ 1^k)=1]| < (p^{QR}+p^R+p^C)$

where PROB[·] is the *privacy* revealing probability, $Adv(\cdot)$ is the trusted-but-curious server, $Ad(\cdot)$ is the polynomial time intermediate adversary, k is the security parameter, p^{QR} is the QRA probability, p^R is the single fair coin toss probability, p^C is some combination identification probability.

Let n bit 1-dimensional matrix database be $\mathcal{DB}= \{b_1,\ b_2,\ ...,\ b_n\}$. Consider S_a,$S_o \subseteq \mathcal{DB} \times \mathcal{DB}$ where S_a is viewed as the subset of ordered pairs $\{\mathcal{B}_1,\ \mathcal{B}_2,\cdots,\ \mathcal{B}_h\}$ where $\mathcal{B}_1= (b_1,\ b_2)$, $\mathcal{B}_2= (b_3,\ b_4)$, and so on till $\mathcal{B}_h= (b_{n\text{-}1},\ b_n)$ and S_o is viewed as the subset of ordered pairs $\{\mathcal{B}_1^{'},\ \mathcal{B}_2^{'},\cdots,\ \mathcal{B}_{h\text{-}1}^{'}\}$ where $\mathcal{B}_1^{'}= (b_2,\ b_4)$, $\mathcal{B}_2^{'}= (b_4,\ b_6)$, and so on till $\mathcal{B}_{h\text{-}1}^{'}= (b_{n\text{-}2},\ b_n)$ and $h=\frac{n}{2}$ is the total number of ordered pairs of S_a. Let the communication bit sets for S_a and S_o be $T_a=\{i : i \in \{0,1\}\}$ and $T_o=\{j : j \in \{0,1\}\}$ respectively with $|T_a|=\frac{n}{2}$-1 and $|T_o|=\frac{n}{2}$-2.

Note that the bits are arranged in S_a and S_o in such a way that second bit of each $\mathcal{B}_i$, $1 \leq i \leq h-1$, is same as first bit of $\mathcal{B}_i^{'}$. All the following subsections use this database (i.e., $\mathcal{DB}$) definition only.

3.1 Building Blocks

Communication bit: It is a special bit (i.e., $hx \in \{0,1\}$) used as a "trapdoor" to inverse the LTDF ($\mathcal{T}$) described in Section 2. If the input α of the LTDF ($\mathcal{T}$) of Section 2 is restricted to $\mathbb{Z}_N^{+1}$, then the value of "jx" of α is always zero. Then, the modified function would be $\mathcal{T}=(\alpha^2 \cdot r^0 \cdot s^{hx} \equiv z \pmod N) \Rightarrow \mathcal{T}^{'}=(\alpha^2 \cdot s^{hx} \equiv z \pmod N)$. Therefore, we have considered this modified function $\mathcal{T}^{'}$ to define a reduced new communication bit function $\mathcal{MT}$: $\mathbb{Z}_N^{+1} \rightarrow \mathbb{Z}_N^{+1}$. For all $\alpha \in \mathbb{Z}_N^{+1}$, the communication bit function is

$$\mathcal{MT}(\alpha) = (\alpha^2 \cdot w^{\delta\%2} \equiv z \pmod N) =< z, hx_\alpha > \tag{1}$$

where $\alpha \in \mathbb{Z}_N^{+1}$, $w \in \overline{Q}_R$, $\delta \in \{0,1,2\}$, the value of "hx_α" is considered as the "communication bit". The value of hx_α is 1 if $\alpha > N/2$ otherwise hx_α is 0. The respective inverse is $\mathcal{MT}^{-1}(z,hx_\alpha)=(\sqrt[0,hx_\alpha]{z \cdot (w^{\delta\%2})^{-1}} \equiv \alpha \pmod N)$.

QRA based Single Bit Encryption (SBE): For all bit $b \in \{0,1\}$, for all random $x, y, \mathcal{PK}_1, \mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$ with $QRP(\mathcal{PK}_1) \neq QRP(\mathcal{PK}_2)$, for all random $\mathcal{PK}_3 \in \mathbb{Z}_N^{-1}$, the single bit encryption $\mathcal{E}_s(b,N,x,y,\mathcal{PK}_1,\mathcal{PK}_2,\mathcal{PK}_3)$ is given in Eq. 2. Each input x,$y \in \mathbb{Z}_N^{+1}$ consists of their respective jx,hx values as described in LTDF. There are four jx,hx combinations (listed in the first column of Eq. 2) possible for any $x, y \in \mathbb{Z}_N^{+1}$. Use the respective pair of equations to encrypt the bit b. For instance, if jx_x=0,hx_x=0 and jx_y=0,hx_y=1, then for all b=0 use the pair of equations defined in second row and second column of Eq. 2; for all b=1

Table 1. Possible residuosity property combinations of ciphertext y when the input $x \in Q_R$.

$a\ b\ (x \cdot \mathcal{PK}_l) \cdot \mathcal{PK}_{l'} \equiv y \pmod N$
$0\ 0\ (x \cdot \mathcal{PK}_1) \cdot \mathcal{PK}_1 \equiv y \in Q_R$
$0\ 1\ (x \cdot \mathcal{PK}_1) \cdot \mathcal{PK}_2 \equiv y \in \overline{Q}_R$
$1\ 0\ (x \cdot \mathcal{PK}_2) \cdot \mathcal{PK}_1 \equiv y \in \overline{Q}_R$
$1\ 1\ (x \cdot \mathcal{PK}_2) \cdot \mathcal{PK}_2 \equiv y \in Q_R$

Table 2. Possible residuosity property combinations of ciphertext y when the input $x \in \overline{Q}_R$.

$a\ b\ (x \cdot \mathcal{PK}_l) \cdot \mathcal{PK}_{l'} \equiv y \pmod N$
$0\ 0\ (x \cdot \mathcal{PK}_2) \cdot \mathcal{PK}_2 \equiv y \in \overline{Q}_R$
$0\ 1\ (x \cdot \mathcal{PK}_2) \cdot \mathcal{PK}_1 \equiv y \in Q_R$
$1\ 0\ (x \cdot \mathcal{PK}_1) \cdot \mathcal{PK}_2 \equiv y \in Q_R$
$1\ 1\ (x \cdot \mathcal{PK}_1) \cdot \mathcal{PK}_1 \equiv y \in \overline{Q}_R$

use second row and third column of Eq. 2.

$$\mathcal{E}_s = \begin{cases} \begin{array}{lll} \textbf{jx,hx} & \textbf{If } b=0 & \textbf{If } b=1 \\ 0,0 \quad x^2 \cdot \mathcal{PK}_1 \equiv c_1 & x^2 \cdot \mathcal{PK}_1 \equiv c_1 \\ 0,0 \quad y^2 \cdot \mathcal{PK}_3 \equiv c_2 & y^2 \cdot \mathcal{PK}_2 \equiv c_2 \end{array} \Big\} \textbf{if } x \leq \mathrm{N}/2, y \leq \mathrm{N}/2 \\ \begin{array}{lll} 0,0 \quad x^2 \cdot \mathcal{PK}_1 \equiv c_1 & x^2 \cdot \mathcal{PK}_3 \equiv c_1 \\ 0,1 \quad y^2 \cdot \mathcal{PK}_1 \equiv c_2 & y^2 \cdot \mathcal{PK}_1 \equiv c_2 \end{array} \Big\} \textbf{if } x \leq \mathrm{N}/2, y > \mathrm{N}/2 \\ \begin{array}{lll} 0,1 \quad x^2 \cdot \mathcal{PK}_3 \equiv c_1 & x^2 \cdot \mathcal{PK}_2 \equiv c_1 \\ 0,0 \quad y^2 \cdot \mathcal{PK}_2 \equiv c_2 & y^2 \cdot \mathcal{PK}_2 \equiv c_2 \end{array} \Big\} \textbf{if } x > \mathrm{N}/2, y \leq \mathrm{N}/2 \\ \begin{array}{lll} 0,1 \quad x^2 \cdot \mathcal{PK}_2 \equiv c_1 & x^2 \cdot \mathcal{PK}_2 \equiv c_1 \\ 0,1 \quad y^2 \cdot \mathcal{PK}_1 \equiv c_2 & y^2 \cdot \mathcal{PK}_3 \equiv c_2 \end{array} \Big\} \textbf{if } x > \mathrm{N}/2, y > \mathrm{N}/2 \end{cases} \tag{2}$$

The decryption of $\mathcal{E}_s$ to get the bit b involves the identification of respective quadratic residuosity properties of the ciphertexts c_1 and c_2 as follows.
Step-1: Identify $QRP(c_1)$ and $QRP(c_2)$. Output the respective b and (jx,hx) combinations of x, y.
Step-2: Find x^2, y^2 using the respective public key inverses. Then, given x^2 and (jx_x, hx_x) values, identify x as described in Eq. 1. Similarly, given y^2 and (jx_y, hx_y) values, identify y as described in Eq. 1.

Axiom 1 *For all RSA composite N=pq where $|p|$=$|q|$=k, $|Q_R|$=$|\overline{Q}_R|$=1/$|\mathbb{Z}_N^{+1}|$.*

Pair-link Encryption (PLE): It is newly constructed quadratic residuosity based encryption method which encrypts two bits at a time and its respective decryption function decrypts only a single bit (Note: PLE is analogous to logical "xor" operation).

For any ordered bit pair $(a, b) \in \mathcal{B}_i$, $i \in [h]$, of S_a or $(a, b) \in \mathcal{B}_j^{'}$, $j \in [h-1]$, of S_o of n bit database $\mathcal{DB}$ where h=$n/2$ and for all the random input $t \in \overline{Q}_R$ and for all the public key $\mathcal{PK}_1$, $\mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$ with $\mathrm{QRP}(\mathcal{PK}_1) \neq \mathrm{QRP}(\mathcal{PK}_2)$, the encryption function $\mathcal{E} : \mathbb{Z}_N^{+1} \to \mathbb{Z}_N^{+1}$ is given as

$$\mathcal{E}((a, b), x = t^{\rho}, \mathcal{PK}_1, \mathcal{PK}_2) = ((x \cdot \mathcal{PK}_l) \cdot \mathcal{PK}_{l'} \equiv y \pmod N) \tag{3}$$

where l,$l^{'} \in [2]$ and $\rho \in \{1, 2\}$.

In order to understand this method, let us look at the encryption tables Table 1,2. For instance, let us consider Table 1 and the input pair a=0, b=1 and the random input $x \in Q_R$. Let $\mathcal{PK}_l \in Q_R$, $\mathcal{PK}_2 \in \overline{Q}_R$. Now, the encryption of a=0 and b=1 is $\mathcal{E}((0,1),x,\mathcal{PK}_1,\mathcal{PK}_2)= ((x \cdot \mathcal{PK}_l)\cdot\mathcal{PK}_{l'} \equiv y \pmod N)$ where l=1, $l^{'}$=2 and the ciphertext y is always a quadratic non-residue. Similarly, the encryption of a=1 and b=1 is $\mathcal{E}((1,1),x,\mathcal{PK}_1,\mathcal{PK}_2)= ((x \cdot \mathcal{PK}_l)\cdot\mathcal{PK}_{l'} \equiv y \pmod N)$ where l=2, $l^{'}$=2 and the ciphertext y is always a quadratic residue and so on.

In order to find the decryption of PLE, one should know the value of the second bit b in advance (bit b acts as an inverse factor) and the quadratic residuosity properties of the ciphertext y and the input x. On identifying the residuosity property of the ciphertext y and given b (Note: there is some means to get the second bit b when this isolated encryption instance is used in combination with other encryption instances in CCE or SWE methods. At this point, assume that the second bit b is given), the decryption of PLE to get the first bit a is calculated as

$$\mathcal{E}^{-1}(y, b, \mathcal{SK}_1, \mathcal{SK}_2) = ((y \cdot \mathcal{SK}_l) \cdot \mathcal{SK}_{l'} \equiv x \pmod N) =< x, a > \tag{4}$$

where l,$l^{'} \in [2]$ and $\mathcal{PK}_l \cdot \mathcal{SK}_l \equiv \mathcal{PK}_{l'} \cdot \mathcal{SK}_{l'} \equiv 1 \pmod N$.

If the quadratic residuosity properties of the ciphertext y and the input x (Note: the quadratic residuosity property of a number should be calculated using the private key p, q) and the second bit b is known then the first bit a can easily be calculated. Initially, using the private key p, q, identify quadratic residuosity property of y. Then, given QRP(y) and b, identify the corresponding first bit a and the public key inverse combinations $\mathcal{SK}_l$, $\mathcal{SK}_{l'}$ for the unique quadratic residuosity property combinations of the ciphertext y (i.e., QRP(y)) and the input x (i.e., QRP(x)). Finally, using the identified public key inverse combinations $\mathcal{SK}_l$, $\mathcal{SK}_{l'}$, l,$l^{'} \in [2]$, get back the input x. For example, if the pair a=0, b=1 is encrypted then the ciphertext y is always a quadratic non-residue i.e., QRP(y)=1 and let the input x is a quadratic residue i.e., QRP(x)=0. Now, it is clear that, for QRP(y)=1, QRP(x)=0 and b=1, a is equal to 0 and l=2, $l^{'}$=1. Finally, get the input x as $(y\cdot\mathcal{SK}_2)\cdot\mathcal{SK}_1 \equiv x \pmod N$. Similarly, if the pair a=1, b=1 is encrypted then the ciphertext y is always a quadratic residue i.e., QRP(y)=0 and let the input x is a quadratic residue i.e., QRP(x)=0. Now, it is clear that, for QRP(y)=0, QRP(x)=0 and b=1, a is equal to 1 and l=1, $l^{'}$=1. Finally, get the input x as $\mathcal{E}^{-1}(y,b,\mathcal{SK}_1,\mathcal{SK}_2)=(y \cdot \mathcal{SK}_1)\cdot\mathcal{SK}_1 \equiv x \pmod N$. Note that, for a single encryption-decryption instance and for a given $\rho \in \{1, 2\}$, use any one of the tables Table 1,2.

Lemma 1. *Let $N \in \{0,1\}^k$ be the RSA composite. Let p^R be a single fair coin toss probability, p^C be a combination selection probability and p^{QR} be a QRA probability. For all given $\mathcal{E}((a, b), \cdot)$ where $a, b, \in \{0, 1\}$, public key (N, $\mathcal{PK}_1 \in \mathbb{Z}_N^{+1}$, $\mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$), for all probabilistic polynomial time intermediate adversary $Ad(\cdot)$, for any random number $x \in \mathbb{Z}_N^{+1}$, for all security parameter k,*

$$\text{PROB}[\mathcal{E}((a, b), x, \mathcal{PK}_1, \mathcal{PK}_2) : Ad(N, \mathcal{PK}_1, \mathcal{PK}_2, \mathcal{E}) = (a, b)] < (p^{QR} + p^R + p^C) \tag{5}$$

Proof. Since the pair-link encryption described in Eq. 3 is based on quadratic residuosity, the adversary has minimum probability equivalent to $p^{QR}= (1/2 + p(k))$ where $p(k)$ is some inverse polynomial in k. Also, for any two equal plaintext bits, the pair-link function in Eq. 3 always produces with the same property ciphertext (Refer the tables Table 1,2). By this approach, the intermediate adversary has additional p^R=1/2 probability to get the correct plaintext along with p^{QR}. In addition to that, the encryption of Eq. 3 uses eight combination tables for any input $x \in \mathbb{Z}_N^{+1}$ and public key $\mathcal{PK}_1$, $\mathcal{PK}_2$. Hence, probability of getting the exact combination is p^C=1/8. Therefore, the total success probability to know the exact plaintext bits (a,b) would always be less than p^{QR}+p^R+p^C.

Axiom 2 *For all $a, b \in \mathbb{Z}_N^{+1}$, the equation $ax \equiv b \ (mod\ N)$ always has a unique solution if $gcd(a, N)$=1.*

Lemma 2. *Every pair-link encryption described in Eq. 3 has unique solution.*

Proof. It is clear from the Axiom 2 that if the *gcd* of participating equation variables w.r.t the modulus is 1, then there exists a unique modular solution. Therefore, for every unique one-to-one mapping function $\mathcal{E}$ (as given in Eq. 3) there exists a unique inverse mapping function $\mathcal{E}^{-1}$ (as given in Eq. 4). In other words, for all the pair (a, b) and the random input $x \in \mathbb{Z}_N^{+1}$ and public key $\mathcal{PK}_1$, $\mathcal{PK}_2$, the encryption of Eq. 3 always maps to unique y since $gcd(\mathcal{PK}_1,N)$=1 and gcd($\mathcal{PK}_2$,N)=1 and $p, q \equiv 3 \pmod 4$. Similarly, for all the given ciphertext y and QRP(x) and the second bit b, the decryption of Eq. 4 always gives the unique bit a and the input x. That means, for any given encryption table i.e., Table 1,2, the encryption $\mathcal{E}$ always maps to the unique property ciphertext (though the value of the ciphertext may be different) and for any given corresponding decryption, the decryption $\mathcal{E}^{-1}$ always maps to the unique bit and the unique input. This implies that Eq. 3 has unique solution.

Connective function ($\mathcal{C}$): Since the individual PLE described in Eq. 3 alone can encrypt only two bits at a time, connect any two successive pair-link encryptions with the aid of the communication bit function $\mathcal{MT}$ of Eq. 1 to encrypt two pair of bits (instead of a single pair). Therefore, either select the input bits $a, b, c, d \in \{0, 1\}$ where $(a, b) \in \mathcal{B}_i$ and $(c, d) \in \mathcal{B}_{i+1}$, with the condition $b \neq c$, $1 \leq i \leq (h\text{-}1)$, from S_a or select the input bits $a, b, c, d \in \{0, 1\}$ where the ordered pairs $(a, b) \in \mathcal{B}_j^{'}$ and $(c, d) \in \mathcal{B}_{j+1}^{'}$, with the condition b=c, $1 \leq j \leq (h\text{-}2)$, from S_o of the n bit database $\mathcal{DB}$.

For all bit pairs $(a, b), (c, d)$ as selected above and input x and the public key $\mathcal{PK}_1$, $\mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$ with QRP($\mathcal{PK}_1$)$\neq$QRP($\mathcal{PK}_2$) and for any two successive pair-link encryptions $\mathcal{E}$ and $\mathcal{F}$ as each described in Eq. 3, the connective function $\mathcal{C}$: $\mathbb{Z}_N^{+1} \rightarrow \mathbb{Z}_N^{+1}$ is given as

$$\mathcal{C}((a, b), (c, d), \mathcal{E}, \mathcal{F}) = \mathcal{F}((c, d), \mathcal{MT}(\mathcal{E}((a, b), x, \mathcal{PK}_1, \mathcal{PK}_2) = y) =< y^2, t >, \mathcal{PK}_1, \mathcal{PK}_2) = z \tag{6}$$

where l,$l^{'} \in [2]$, t is equal to 1 if $y > N/2$ otherwise t is equal to 0. We treat this "t" as "communication bit" equivalent to the "hx" value described in communication bit function $\mathcal{MT}$.

In the connective function of Eq. 6, the preceding pair-link encryption $\mathcal{E}$ receives the ordered pair of bits (a, b), the input x, the public key $\mathcal{PK}_1$, $\mathcal{PK}_2$ with QRP($\mathcal{PK}_1$)$\neq$QRP($\mathcal{PK}_2$) and encrypts (a, b) and generates the ciphertext y. Further, the communication bit function $\mathcal{MT}$ receives the ciphertext y as input and produces the ciphertext y^2 and the communication bit t where t=1 if $y > N/2$ otherwise t=0. Finally, the succeeding pair-link encryption $\mathcal{F}$ receives the ordered pair of bits (c, d), the ciphertext y^2 as input, the public key $\mathcal{PK}_1$, $\mathcal{PK}_2$ and encrypts (c, d) and generates the final ciphertext z. Therefore, any two successive pair-link encryptions connected using connective function $\mathcal{C}$ always encrypt four bits and produce one communication bit in between them.
Types of connection: We define two types of successive pair-link encryption connections as follows.

- *Criss-cross*: In this, every successive ordered pairs $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$, $1 \leq i \leq$ (h-1), of S_a are encrypted using connective function. Similarly, every successive ordered pairs $\mathcal{B}_j^{'}$ and $\mathcal{B}_{j+1}^{'}$, $1 \leq j \leq$ (h-2), of S_o are encrypted using connective function ($\mathcal{C}$).
- *Snake-walk*: Let $S_f \subseteq \mathcal{DB} \times \mathcal{DB}$ where S_f is viewed as a set of ordered pairs $\{\mathcal{B}_1^{''}, \mathcal{B}_2^{''}, \cdots, \mathcal{B}_{h\text{-}1}^{''}\}$ where $\mathcal{B}_1^{''}$= (b_2, b_3), $\mathcal{B}_2^{''}$= (b_4, b_5), and so on till $\mathcal{B}_{h\text{-}1}^{''}$= ($b_{n\text{-}2}$, $b_{n\text{-}1}$). In this, every successive pair $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$, $1 \leq i \leq$ (h-1), of S_a are encrypted using connective function. Similarly, every successive pairs $\mathcal{B}_j^{''}$ and $\mathcal{B}_{j+1}^{''}$, $1 \leq j \leq$ (h-2), of S_f are encrypted using connective function ($\mathcal{C}$).

In order to decrypt the connective function of Eq. 6, the seconds bits d, b that were encrypted using $\mathcal{F}$ and $\mathcal{E}$ are essential along with the quadratic residuosity properties of z and x. How to get these second bits ? or Who will provide these second bits ?

By careful observation, in the *criss-cross* connection type, it is clear that the decryption of every connective function involving encryption of successive pairs $\mathcal{B}_j^{'}$ and $\mathcal{B}_{j+1}^{'}$, $1 \leq j \leq$ (h-2), of S_o provides the second bits for the decryption of every connective function involving encryption of successive pairs $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$, $1 \leq i \leq$ (h-1), of S_a (since every first bit of $\mathcal{B}_j^{'}$ and $\mathcal{B}_{j+1}^{'}$ are same as every second bit of $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$). Similarly, in the *snake-walk* connection type, it is clear that the decryption of every connective function involving encryption of successive pairs $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$, $1 \leq i \leq$ (h-1), of S_a provides the second bits for the decryption of every connective function involving encryption of successive pairs $\mathcal{B}_j^{''}$ and $\mathcal{B}_{j+1}^{''}$, $1 \leq j \leq$ (h-2), of S_f since every first bit of $\mathcal{B}_i$ and $\mathcal{B}_{i+1}$ are same as every second bit of $\mathcal{B}_j^{''}$ and $\mathcal{B}_{j+1}^{''}$.

Definition 3 *Chain of successive connective function ($\mathcal{CHAIN}$)* It is the chain of successive connective functions of the form $\mathcal{CHAIN}(N,\ S,\ \alpha,\ \mathcal{PK}_1,\ \mathcal{PK}_2)$= $([\xrightarrow{\alpha} C_1 \Rightarrow o_1] \xrightarrow{o_1} [C_2 \Rightarrow o_2] \xrightarrow{o_2} \cdot\cdot \xrightarrow{o_{g\text{-}2}} [C_{g\text{-}1} \Rightarrow o_{g\text{-}1}] \xrightarrow{o_{g\text{-}1}} [C_g \Rightarrow o_g])$=$< o_g, \mathcal{TB} >$ where $1 \leq g \leq h$, h=$n/2$, $(\alpha, \mathcal{PK}_1, \mathcal{PK}_2) \in \mathbb{Z}_N^{+1}$, $S \subseteq \mathcal{DB} \times \mathcal{DB}$ and o_g is the

final output ciphertext and $\mathcal{TB}$ is the communication bit set and each connective function $\mathcal{C}$ is drawn from Eq. 6. Let $\mathcal{CHAIN}^{-1}(o_g,\mathcal{TB}, (p,q))$ is the respective inverse chain.
Remark: For the CCE encryption, there are two concurrently executing chains $\mathcal{CHAIN}_1(N,S_a,\alpha,\mathcal{PK}_1,\mathcal{PK}_2)$ with g=h-1 and $\mathcal{CHAIN}_2(N,S_o,\alpha,\ \mathcal{PK}_1,\ \mathcal{PK}_2)$ with g=h-2 in which the subset S_a is encrypted using $\mathcal{CHAIN}_1$ and the subset S_o is encrypted using $\mathcal{CHAIN}_2$ resulting in the generation of g number of communication bits from each chain along with each chain ciphertexts. Similarly, for the SWE type of encryption, there are two concurrently executing chains $\mathcal{CHAIN}_1(N,S_a,\alpha,\mathcal{PK}_1,\mathcal{PK}_2)$ with g=h and $\mathcal{CHAIN}_2(N,S_f,\alpha,\ \mathcal{PK}_1,\ \mathcal{PK}_2)$ with g=h-1 in which the subset S_a is encrypted using $\mathcal{CHAIN}_1$ and the subset S_f is encrypted using $\mathcal{CHAIN}_2$ resulting in the generation of g number of communication bits from each chain along with each chain ciphertexts.

3.2 Proposed sitPIR Scheme

In order to generate the response from the server, the main trick here is to execute two chain of successive connective functions (individual connective function $\mathcal{C}$ is described in Subsection 3.1 and the chain of successive connective functions is described in Definition 3) in parallel on the database and produce the respective ciphertexts. Also, encrypt the last database bit using QRA based single bit encryption (SBE) of Section 3.1 and produce the final ciphertexts. Consider the criss-cross type of encryption for instance. The detailed description of the algorithms is as follows.

– **Query Generation (QG)**: Let $N \xleftarrow{R} \{0,1\}^k$. User $\mathcal{U}_{pir}$ sends information-theoretic query $\mathcal{Q}$=(α, N, $\mathcal{PK}_1$, $\mathcal{PK}_2$, $\mathcal{PK}_3$) to the server where $\mathcal{PK}_1$, $\mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$ with QRP($\mathcal{PK}_1$)$\neq$QRP($\mathcal{PK}_2$), $\mathcal{PK}_3 \in \mathbb{Z}_N^{-1}$ and $\alpha \xleftarrow{R} \mathbb{Z}_N^{+1}$.
– **Response Creation (RC)**: Server $\mathcal{S}_{pir}$ generates the response $\mathcal{R}$ consisting of two ciphertexts and two communication bit sets as follows.

Initially, using the query $\mathcal{Q}$, server executes two parallel chain of successive connective functions $\mathcal{CHAIN}_1$ and $\mathcal{CHAIN}_2$ (each chain is described in Definition 3) on either using CCE or SWE type of the database and produces respective chain ciphertexts β_1, β_2 and respective communication bit sets T_a and T_o as follows. Consider CCE type for instance. All the ordered pairs of the subset S_a are encrypted using $\mathcal{CHAIN}_1$ as

$$\begin{aligned}\mathcal{CHAIN}_1(N,S_a,\alpha,\mathcal{PK}_1,\mathcal{PK}_2) &= \mathcal{C}_i(N,\mathcal{MT}(\mathcal{C}_{i-1}),\mathcal{PK}_1,\mathcal{PK}_2)\\ &=<\beta_1,(T_a=(t_1,\cdot\cdot,t_{i-1}))>\\ &=<\beta_1,T_a>\end{aligned} \tag{7}$$

where δ=ρ=2, $i \in [h,2]$,h=$n/2$, β_1 is the output ciphertext generated from $\mathcal{CHAIN}_1$, T_a is the communication bit set with (h-1) number of communication bits and $\mathcal{C}_1((b_1,b_2),(b_3,b_4),\mathcal{E}_1,\mathcal{E}_2)$=$\mathcal{E}_2((b_3,b_4),\mathcal{MT}(\mathcal{E}_1((b_1,b_2),\alpha,\mathcal{PK}_1,\mathcal{PK}_2)=y)=<y^2,\ t_1>,\ \mathcal{PK}_1,\mathcal{PK}_2)$.

Similarly, all the ordered pairs of the subset S_o are encrypted using $\mathcal{CHAIN}_2$ as

$$\begin{aligned}\mathcal{CHAIN}_2(N, S_o, \alpha, \mathcal{PK}_1, \mathcal{PK}_2) &= \mathcal{C}_i(N, \mathcal{MT}(\mathcal{C}_{i-1}), \mathcal{PK}_1, \mathcal{PK}_2) \\ &=< \beta_2, (T_o = (t_1^{'}, \cdot\cdot, t_{i-1}^{'})) > \\ &=< \beta_2, T_o >\end{aligned} \tag{8}$$

where $i \in [h\text{-}1, 2]$,h=$n/2$,β_2 is the output ciphertext generated from $\mathcal{CHAIN}_2$, T_o is the communication bit set with (h-2) number of communication bits and $\mathcal{C}_1((b_2,b_4),(b_4,b_6),\mathcal{E}_1,\mathcal{E}_2)$=$\mathcal{E}_2((b_4,b_6),\mathcal{MT}(\mathcal{E}_1((b_2,b_4),\alpha,\mathcal{PK}_1,\mathcal{PK}_2) = y) =< y^2,t_1^{'} >$, $\mathcal{PK}_1$,$\mathcal{PK}_2$).

It is evident that both the chains $\mathcal{CHAIN}_1$ and $\mathcal{CHAIN}_2$ interlock the database bits (we call this type of encryption as "*criss-cross encryption*" and the cipher generated from it as criss-cross cipher alternative to substitution or transposition ciphers) and hence, all the ordered pairs of subsets S_a and S_o should be retrieved alternatively using the respective inverse chains $\mathcal{CHAIN}_1^{-1}$ and $\mathcal{CHAIN}_2^{-1}$. That means, every second bit of each ordered pair of S_a is encrypted as a first bit of each pair of S_o. Hence, during retrieval, it is impossible to retrieve the required bit(s) of the subset S_a or S_o alone without the aid of other inverse chain.

Further, the last bit b_n is encrypted using the single bit encryption SBE as $\mathcal{E}_s(b_n,N,\beta_1,\beta_2,\mathcal{PK}_1,\mathcal{PK}_2,\mathcal{PK}_3)$=$(\gamma_1,\gamma_2)$. Finally, the PIR response $\mathcal{R}$ is generated as $\mathcal{R}$=$\{\mathcal{C}_1$=$(\gamma_1,T_a),\mathcal{C}_2$=$(\gamma_2,T_o)\}$. Therefore, for the whole database, there are two constant k size ciphertexts and $(2h-3)$ number of communication bits generated in total. This response $\mathcal{R}$ is sent back to the user.

– **Interest Extraction (IE)**: Using the response $\mathcal{R}$ and the private key (p, q), user $\mathcal{U}_{pir}$ privately reads the required bit of the database $\mathcal{DB}$ as follows.

Initially, using the ciphertext (γ_1,γ_2), find the last bit b_n as $\mathcal{E}_s(\gamma_1,\gamma_2)$=$b_n$. Since both the chains were adopted criss-cross encryption during response creation on the server, exact reverse order should be maintained to get the required bit using the obtained last bit b_n and chain specific ciphertexts β_1, β_2 and T_a,T_o.

It is intuitive that the last bit b_n of the database $\mathcal{DB}$ is always same as the second bit of $\mathcal{B}_h \in S_a$ and $\mathcal{B}_{h\text{-}1}^{'} \in S_o$. Since both the chains $\mathcal{CHAIN}_1$ and $\mathcal{CHAIN}_2$ have adopted criss-cross encryption, it is also clear that the first bit of each $\mathcal{B}_i^{'} \in S_o$ is always equal to second bit of each $\mathcal{B}_i \in S_a$ where h-1 $\geq i \geq$ 1. Hence, find the first bits of $\mathcal{B}_h \in S_a$ and $\mathcal{B}_{h\text{-}1}^{'} \in S_o$ by inverting respective chains $\mathcal{CHAIN}_1$ and $\mathcal{CHAIN}_2$ and continue the inverse process till the required bit of interest.

3.3 A Toy Example:

Let us consider N=133, p=19, q=7 and database $\mathcal{DB}$=$\{$1,1,0,0, 1,1,1,1$\}$ where $|\mathcal{DB}|$=n=8. Therefore, S_a=$\{$(1,1), (0,0), (1,1), (1,1)$\}$ and S_o=$\{$(1,0), (0,1), (1,1)$\}$. Let α=25, $\mathcal{PK}_1$=44, $\mathcal{PK}_2$=48, $\mathcal{PK}_3$=15. Let us assume that the user is interested in b_5. An illustrative example is given in Table 3.

Table 3. An illustrative example of response creation (RC) and interest extraction (IE) algorithms of the proposed sitPIR scheme.

Response Creation (RC)			
	Step-1		*Step-2*
	$\mathcal{CHAIN}_1(N, S_a, \alpha, \mathcal{PK}_1, \mathcal{PK}_2)$	$\mathcal{CHAIN}_2(N, S_o, \alpha, \mathcal{PK}_1, \mathcal{PK}_2)$	$\mathcal{E}_s(\cdot)$
1	$\mathcal{E}(1,1,25,133,44,48)$=11	$\mathcal{E}(1,0,25,133,44,48)$=132	$92^2 \cdot 48 \equiv 90$
	$\mathcal{MT}(11) =< 121, 0 >$	$\mathcal{MT}(132) =< 1, 1 >$	$102^2 \cdot 15 \equiv 51$
2	$\mathcal{E}(0,0,121,133,44,48)$=43	$\mathcal{E}(0,1,1,133,44,48)$=117	
	$\mathcal{MT}(43) =< 120, 0 >$	$\mathcal{MT}(117) =< 123, 1 >$	
3	$\mathcal{E}(1,1,120,133,44,48)$=106	$\mathcal{E}(1,1,123,133,44,48)$=102	
	$\mathcal{MT}(106) =< 64, 1 >$		
4	$\mathcal{E}(1,1,64,133,44,48)$=92		
	β_1=92, T_a=(0,0,1)	β_2=102, T_0=(1,1)	γ_1=90,γ_2=51
Therefore, $\mathcal{C}_1$=<90,(0,0,1)>, $\mathcal{C}_2$=<51,(1,1)>			
Interest Extraction (IE)			
	Step-2		*Step-1*
	$\mathcal{CHAIN}_1^{-1}(\beta_1, T_a, p, q)$	$\mathcal{CHAIN}_2^{-1}(\beta_2, T_o, p, q)$	$\mathcal{E}_s^{-1}(\cdot)$
1	$\mathcal{E}^{-1}(1,92,19,7,130,97)$=<64,1>	$\mathcal{E}^{-1}(1,102,19,7,130,97)$=<123,1>	$90 \in \overline{Q}_R$
	$\mathcal{MT}^{-1}(64,0,1) = 106$		$51 \in \mathbb{Z}_N^{-1}$
2	$\mathcal{E}^{-1}(1,106,19,7,130,97)$=<120,1>		
	b_5=1		b_n=1

3.4 Security Proofs

Lemma 3. *For all pair-link encryption ($\mathcal{E}$) described in Lemma 2, the input $x \in \mathbb{Z}_N^{+1}$ is identically distributed over Q_R and $\overline{Q}_R$ (or identically distributed over $\mathbb{Z}_N^{+1}$).*

Proof. Case-1: Let us consider $x \in Q_R$ and the public key $\mathcal{PK}_1$, $\mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$ where $\mathcal{PK}_1$ and $\mathcal{PK}_2$ belong to different subsets either Q_R or $\overline{Q}_R$ (i.e.,($\mathcal{PK}_1 \in Q_R$,$\mathcal{PK}_2 \in \overline{Q}_R$) or ($\mathcal{PK}_1 \in \overline{Q}_R$,$\mathcal{PK}_2 \in Q_R$)) from Table 1,2. For all the pair $a, b \in \{0,1\}$ with a=b, $\mathcal{E}$ successfully generates a unique quadratic residue ciphertext $y \in Q_R$. Similarly, for all the pair $a, b \in \{0,1\}$ with $a \neq b$, $\mathcal{E}$ successfully generates a unique quadratic non residue ciphertext $y \in \overline{Q}_R$.

Case-2: Let us consider $x \in \overline{Q}_R$ and the public key $\mathcal{PK}_1$,$\mathcal{PK}_2 \in \mathbb{Z}_N^{+1}$ where $\mathcal{PK}_1$ and $\mathcal{PK}_2$ belong to different subsets either Q_R or $\overline{Q}_R$ (i.e., ($\mathcal{PK}_1 \in Q_R$,$\mathcal{PK}_2 \in \overline{Q}_R$) or ($\mathcal{PK}_1 \in \overline{Q}_R$,$\mathcal{PK}_2 \in Q_R$)). For all the pair $a, b \in \{0,1\}$ with a=b, $\mathcal{E}$ successfully generates a unique quadratic non residue ciphertext $y \in \overline{Q}_R$. Similarly, for all the pair $a, b \in \{0,1\}$ with $a \neq b$, $\mathcal{E}$ successfully generates a unique quadratic residue ciphertext $y \in Q_R$.

Therefore, for all $x \in \mathbb{Z}_N^{+1}$ (whether $x \in Q_R$ or $x \in \overline{Q}_R$) and for all the input pair $a, b \in \{0,1\}$, the ciphertext generated from $\mathcal{E}$ encryption function is always identically distributed over $\mathbb{Z}_N^{+1}$. Therefore, it is now intuitive that the input x drawn from $\mathbb{Z}_N^{+1}$ is "identically distributed" over Q_R and $\overline{Q}_R$.

Theorem 1 *Any two randomly generated PIR queries from the proposed sitPIR scheme as described in Definition 1 are identically distributed and hence are information-theoretically indistinguishable.*

Proof. From the response creation algorithm RC of the proposed PIR scheme described in Section 3.2, it is clear that each response creation process involves the execution of two parallel chains of successive connective functions and input number to each connective function is always identically distributed over $\mathbb{Z}_N^{+1}$ as described in Lemma 3. Since the input of each $\mathcal{E}$ function is identically distributed over $\mathbb{Z}_N^{+1}$, any two randomly generated PIR queries $\mathcal{Q}_i$ and $\mathcal{Q}_j$, $i, j \in [n]$, with the respective inputs (say) $r \in \mathbb{Z}_N^{+1}$ and $s \in \mathbb{Z}_N^{+1}$ are always identically distributed. Since the queries $\mathcal{Q}_i$ and $\mathcal{Q}_j$ are identically distributed over $\mathbb{Z}_N^{+1}$,

$$\begin{aligned} \text{PROB}[(\mathcal{Q}_i, sk) \xleftarrow{R} QF(1^k) : Adv(n, \mathcal{Q}_i, 1^k) = 1] = \text{PROB}[(\mathcal{Q}_j, sk) \\ \xleftarrow{R} QF(1^k) : Adv(n, \mathcal{Q}_j, 1^k) = 1] \end{aligned} \tag{9}$$

Hence any two randomly selected queries $\mathcal{Q}_i$ and $\mathcal{Q}_j$ from query generation algorithm are always independent to each other and consist of "identically distributed" input numbers.

If the queries are identically distributed, then the privacy leak through the mutual information is always zero. Therefore, let any two independent random variables X and Y be $[(\mathcal{Q}_i, sk) \xleftarrow{R} QF(1^k) : Adv(n, \mathcal{Q}_i, 1^k)=1]$ and $[(\mathcal{Q}_j, sk) \xleftarrow{R} QF(1^k) : Adv(n, \mathcal{Q}_j, 1^k)=1]$ respectively. Intuitively $\text{PROB}(XY)=\text{PROB}(X, Y) = \text{PROB}(X) \cdot \text{PROB}(Y)=\text{PROB}(YX)$. Then the conditional distribution of X and Y is calculated as

$$\begin{aligned} \textbf{PROB}\boldsymbol{(X|\, Y)} &= \frac{\text{PROB}(XY)}{\text{PROB}(Y)} = \frac{\text{PROB}(X) \cdot \text{PROB}(Y)}{\text{PROB}(Y)} = \text{PROB}(X) \\ \textbf{PROB}\boldsymbol{(Y|X)} &= \frac{\text{PROB}(YX)}{\text{PROB}(X)} = \frac{\text{PROB}(Y) \cdot \text{PROB}(X)}{\text{PROB}(X)} = \text{PROB}(Y) \end{aligned} \tag{10}$$

Then, the mutual information of X and Y is calculated as

$$\boldsymbol{I(X, Y)} = \sum_X \sum_Y \text{PROB}(X, Y)\ log\ \frac{\text{PROB}(X, Y)}{\text{PROB}(X) \cdot \text{PROB}(Y)} = 0 = \boldsymbol{I(X, Y)} \tag{11}$$

Intuitively, X and Y are information-theoretically indistinguishable. Therefore, all such queries exhibit *perfect privacy* i.e, leaks no information about the user interest on the curious server side.

Theorem 2 *For all the single database information-theoretically indistinguishable PIR (sitPIR) scheme defined in Definition 1, the server communication cost is always guaranteed to be $\mathcal{O}(o(n)+2\ log\ N)$ where $(2\ log\ N)$ is the fixed size chain specific ciphertexts.*

Proof. By referring the Eq. 7 and Eq. 8, it is clear that the PIR response creation involves two chain of successive connective functions $\mathcal{CHAIN}_1$ and $\mathcal{CHAIN}_2$. There are $(h-1)$ number of connective functions used in $\mathcal{CHAIN}_1$ and each connective function generates one communication bit. Therefore, there are $(h-1)$ number of communication bits generated from $\mathcal{CHAIN}_1$ where $h{=}n/2$. Similarly, there are $(h{-}2)$ number of connective functions used in $\mathcal{CHAIN}_2$ and each connective function generates one communication bit. Therefore, there are $(h-2)$ number of communication bits generated from $\mathcal{CHAIN}_2$. In total, considering both $\mathcal{CHAIN}_1$ and $\mathcal{CHAIN}_2$, there are $(h-1){+}(h-2){=}2h-3 \Rightarrow (2\cdot n/2)-3 \Rightarrow (n-3)$ number of communication bits generated from the database which is clearly less than the database size i.e., $o(n)$. Also, there are two fixed $log\ N$ size chain ciphertexts β_1, β_2. The overall server communication would be $(n-3{+}2\ \log\ N)$ which is slightly greater than the trivial communication (without any optimization). But, the scheme will achieve non-trivial communication when $((o(n){+}2\ \log\ N)/n){=}0$ for all $c_0 > c$ and $n{=}2^{c_0}$ where c is an integer constant.

Correctness proof: When the underlying standard quadratic residuosity product is correct and the LTDF function of Freeman et.al [13] is successfully invertible and the communication bit sets T_a, T_o and the ciphertexts set $\mathcal{R}$ sent from the server are unchanged during transmission, the proposed *sitPIR* scheme always generates the required bit of the database. Therefore, $\forall z \in [n]$, for all the security parameter k ($\forall k \in \mathbb{N}$), for all the RSA composite N, for the database $\mathcal{DB}$,

$$IE\big((\mathcal{R}, sk) : \mathcal{R} \leftarrow RC(\mathcal{Q}, \mathcal{DB}, n, 1^k), (\mathcal{Q}, sk) \xleftarrow{R} QF(1^k)\big) = b_z$$

Proof. By the Lemma 2, it is clear that each pair-link function of Eq. 3 has unique solution. That means, for all the given ciphertext, the inverse pair-link function always produces the unique plaintext. It is intuitive that each chain described in Definition 3 is composed of combination of pair-link functions described in Eq. 3 and LTDFs. Since the underlying components produces, unique solution, the chain also produces the unique solution. Additionally, the crisscross encryption ensures that one chain ($\mathcal{CHAIN}_2$) supplies the required bits to the other chain ($\mathcal{CHAIN}_1$) during retrieval process. Therefore, for all the given ciphertext and communication bits, $\mathcal{CHAIN}_2$ always gives the unique plaintext bits of S_o. For all the given ciphertext and communication bits and the bits supplied by $\mathcal{CHAIN}_2$, $\mathcal{CHAIN}_1$ always gives the unique plaintext bits of S_a. Hence, for all the given response and the private key, the interest extraction (IE) algorithm always produces the required bit of interest.

3.5 Performance

PRIVACY: Since the proposed scheme generates information-theoretic queries, privacy is evenly distributed over $\mathbb{Z}_N^{+1}$. This information-theoretic query makes the curious server to achieve only fair coin toss probability to reveal user privacy. One of the greatest advantages of the proposed scheme is that the *data privacy*

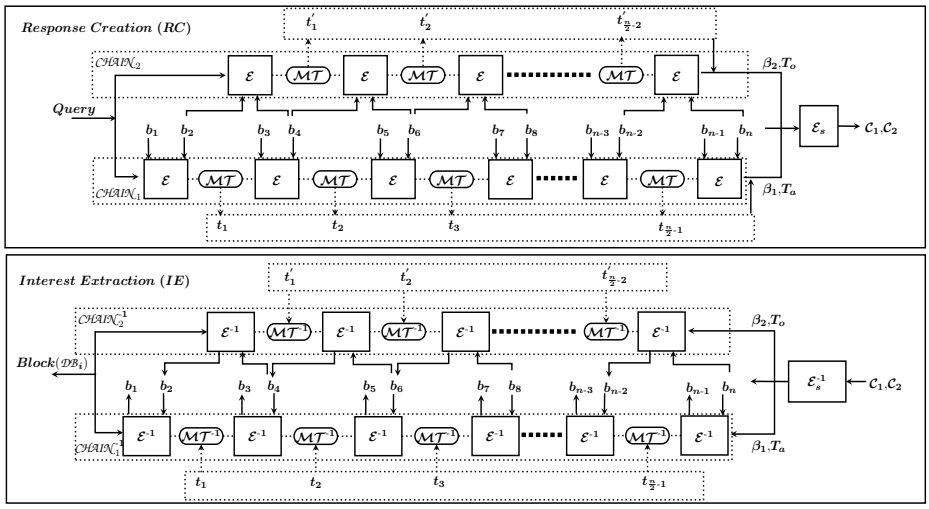

Fig. 1. A single block response creation (RC) and interest extraction (IE) algorithm execution for the proposed sitPBR scheme.

level can be adjusted from (p^{QR}+p^R) to (p^{QR}+p^R+p^C).

COMMUNICATION: For the given database of size n, the proposed scheme generates $\mathcal{O}(5 \log N)$ number of user query bits, generates $\mathcal{O}(n - 3 + 2\log N)$ server response bits. If the caching is enabled by storing all the communication bits generated during response creation, then the succeeding PIR invocations generate only constant size response (i.e., 2 log N bits). Also, note that the non-trivial communication can be achieved when $((o(n)+2 \log N)/n)=0$ for all $c_0 > c$ and $n=2^{c_0}$ where c is an integer constant. For example, if k=2048, the non-trivial communication can be achieved for all c_0=18. Similarly, if k=4096, the non-trivial communication can be achieved for all c_0=20.

COMPUTATION: In the proposed scheme, server executes $\mathcal{O}((3n/2) - 1)$ number of modular multiplications from $\mathcal{CHAIN}_1$ function, $\mathcal{O}((3n/2) - 4)$ number of modular multiplications from $\mathcal{CHAIN}_2$ function and two modular multiplications. User executes minimum two modular multiplications and maximum $\mathcal{O}(3n-5)+2$ number of modular inverse multiplications. In the *criss-cross* method, response creation can be executed with two parallel sub-processes (in which each sub-process executes each chain in parallel) and interest extraction cannot be assigned to sub-processes due to the dependency of one chain on the other. In the *snake-walk* method, both response creation and interest extraction processes can be assigned to two sub-processes.

3.6 A Single Database Information-theoretic Private Block Retrieval (sitPBR)

The proposed sitPIR scheme is easily extended to sitPBR scheme as follows. Let a two dimensional matrix $n= uv$ bit database $\mathcal{D}=\{\mathcal{DB}_1, \cdot\cdot, \mathcal{DB}_u\}$ where $|\mathcal{DB}_i|=v$,

$i \in [1, u]$. The QG algorithm generates identically distributed random queries $\{\mathcal{Q}_1, \cdot\cdot, \mathcal{Q}_u\}$ and the RC algorithm generates $\{\mathcal{R}_1, \cdot\cdot, \mathcal{R}_u\}$ responses. Finally, IE algorithm retrieves the specific block $j \in [u]$ by selecting the respective response $\mathcal{R}_j$ and private key (p, q). The detailed response creation and interest extraction execution for a single database block is given in Fig. 1. Without extra effort, it is evident that the integrity of the response sent by the server is verified when the IE algorithm produces the same residue which was sent in the query.

4 Conclusion

We have successfully constructed the single database information-theoretic PIR scheme using information-theoretic queries to preserve *user privacy* and quadratic residuosity assumption to preserve *data privacy*. The newly constructed *pair-link encryption* and the *criss-cross* and *snake-walk* methods of PIR encryptions using $\mathcal{CHAIN}_1$, $\mathcal{CHAIN}_2$ in RC algorithm together support the information-theoretic single database PIR solution. Even though the proposed scheme fully supports *perfect privacy*, for practical large database applications, it is required to reach reasonable communication cost. Hence, the proposed scheme is only the stepping stone and can further be modified to attain efficient communication cost using pre-processing techniques. There are several additional open problems like considering bandwidth utilization, robustness, fault-tolerance etc. in a single database information-theoretic PIR and among them the construction of communication efficient perfect privacy preserving single database PIR solution for privacy critical applications is still an open problem.

References

1. Carlos Aguilar-Melchor, Joris Barrier, Laurent Fousse, and Marc-Olivier Killijian. Xpir: Private information retrieval for everyone. Cryptology ePrint Archive, Report 2014/1025, 2014. `https://eprint.iacr.org/2014/1025`.
2. Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *Journal of Computer and System Sciences*, 71(2):213 – 247, 2005.
3. Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers computation in private information retrieval:PIR with preprocessing. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 55–73, 2000.
4. Amos Beimel and Yoav Stahl. Robust information-theoretic private information retrieval. *Journal of Cryptology*, 20(3):295–321, 2007.
5. Chor Benny, Gilboa Niv, and Naor Moni. Private information retrieval by keywords. Cryptology ePrint Archive, Report 1998/003, 1998. `http://eprint.iacr.org/1998/003`.
6. Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *Proce. of 17th Theory and Application of Cryptographic Techniques*, EUROCRYPT'99, pages 402–414. Springer-Verlag, 1999.

7. Ran Canetti, Justin Holmgren, and Silas Richelson. Towards doubly efficient private information retrieval. Cryptology ePrint Archive, Report 2017/568, 2017. `https://eprint.iacr.org/2017/568`.
8. Amit Chakrabarti and Anna Shubina. Nearly private information retrieval. In *MFCS 2007*, pages 383–393, 2007.
9. Yan-Cheng Chang. *Single Database Private Information Retrieval with Logarithmic Communication*, pages 50–61. Springer, 2004.
10. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proc. of the 36th FOCS*, FOCS '95, pages 41–50. IEEE Computer Society, 1995.
11. Benny Chor and Niv Gilboa. Computationally private information retrieval (extended abstract). In *Proc. of 29th STOC*, STOC '97, pages 304–313. ACM, 1997.
12. Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.
13. David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. Cryptology ePrint Archive, Report 2009/590, 2009. `http://eprint.iacr.org/2009/590`.
14. Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In *Proc. of 32nd ICALP*, ICALP'05, pages 803–815. Springer-Verlag, 2005.
15. Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *STOC '98*, pages 151–160. ACM, 1998.
16. Ian Goldberg. Improving the robustness of private information retrieval. In *IEEE Symposium on Security and Privacy*, pages 131 – 148, 2007.
17. Jens Groth, Aggelos Kiayias, and Helger Lipmaa. Multi-query computationally-private information retrieval with constant communication rate. In *Proc. of 13th PKC*, PKC'10, pages 107–123. Springer-Verlag, 2010.
18. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *Proc. of 47th FOCS*, FOCS '06, pages 239–248. IEEE Computer Society, 2006.
19. E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proc of 38th FOCS*, FOCS '97, pages 364–. IEEE Computer Society, 1997.
20. Eyal Kushilevitz and Rafail Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *Proc. of 19th Theory and Application of Cryptographic Techniques*, EUROCRYPT'00, pages 104–121. Springer-Verlag, 2000.
21. Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In *Proc. of 8th ISC*, ISC'05, pages 314–328. Springer-Verlag, 2005.
22. Helger Lipmaa. First cpir protocol with data-dependent computation. In *Proc. of 12th Information Security and Cryptology*, ICISC'09, pages 193–210. Springer-Verlag, 2010.
23. Tianren Liu and Vinod Vaikuntanathan. On basing private information retrieval on np-hardness. In *TCC 2016-A*, pages 372–386, 2016.
24. Carlos AGUILAR MELCHOR and Philippe GABORIT. A lattice-based computationally-efficient private information retrieval protocol, 2007.
25. Jonathan Trostle and Andy Parrish. Efficient computationally private information retrieval from anonymity or trapdoor groups. In *Proc. of 13th ISC*, ISC'10, pages 114–128. Springer-Verlag, 2011.