Chapter 6
A Novel Perfect Privacy PIR Scheme for Privacy Critical Applications

Radhakrishna Bhat and N. R. Sunitha

Abstract Majority of the business vendors have incorporated the policy-driven privacy setting that greatly disappoints the end user or customer even though the protocol-based privacy assurance is strongly expected. The reason behind this major disagreement between the vendor and the customer is due to the business survival necessity or the business expansion for the vendor and the mandatory technology adoption or the business monopoly creation for the customer. In order to cope up with the exponentially growing business needs, both vendor and customer have to agree upon the protocol-based privacy setting. Therefore, we have proposed a new generic (i.e., applicable for both client-server and peer-to-peer) perfect privacy preserving information retrieval protocol using the concept of Private Information Retrieval (PIR).

More interestingly, we have overcome the trivial solution of downloading the entire database by achieving $o(n)$ communication cost by introducing a new perfect privacy preserving single database private information retrieval for privacy critical applications using quadratic residuosity as the underlying *data privacy* primitive. Finally, we have concluded by claiming a generic scheme suitable for privacy critical applications.

Keywords Perfect privacy · Private information retrieval · PIR · Quadratic residuosity · Privacy critical applications

6.1 Introduction

The drawback of most of the privacy-enabled retrieval techniques of the server is that they either adopt information theory-based information-theoretic or cryptographic assumption-based computationally bounded privacy techniques which

R. Bhat (✉) · N. R. Sunitha
Siddaganga Institute of Technology, Tumakuru, Karnataka, India

N. S. V. Rao et al. (eds.), *Proceedings of International Symposium on Sensor Networks, Systems and Security*, https://doi.org/10.1007/978-3-319-75683-7_6

are assuring only partial privacy. But the user assumes that his privacy is always guaranteed by the other party. This serious monopoly move by the server leads to several problems.

Scenario-1: Let us consider that all the patients pertaining to a disease of some region have uploaded all their health cards in a plain or encoded format on a server by considering policy-driven privacy assurance.

The server has several possible moves. What if the server shares the stored patient information with other healthcare industries that are eager to know the disease count there by producing more products? What if the server shares the information with other regional bodies which are already tied up for food and nutrition exchange policies?

What is the mere negative consequence? The healthcare industries may pressurize the government body to exchange such food that creates sufficient malnutrition or set a customized health boundary point so that more patients should be included in that boundary. In turn, analytics-enabled pharmaceutical industries may advertise their products so that the physicians should refer to the same products. If this business cycle continues, then at some point in time, all people will become patients.

Scenario-2: Let the public database maintain all the information particular to its domain (e.g., search engines, social media, multimedia, patent, etc.). Let the authenticated user search or retrieve the subset of information from the database to which he is subscribed to. What happens when the analytics-enabled server tracks all the search or retrieval sequences of a particular user or a group of users related to a particular domain? What if the server shares its analytical results with user's business opponent?

Scenario-3: Let us consider that a severe war is happening between two rivals on the war field. What if the global positioning system (GPS) server tracks and shares one of the opponent's livestream information to others?

Scenario-4: Suppose if any two peer devices like two military commanders want to share secret information securely and privately through insecure communication channel or through a mediator? Also, if any two end devices want to communicate with end-to-end encryption enabled like "private chat"? What if the mediator or the third party reveals the communication information?

Perfect Privacy Solution: To overcome the above problems, the only solution is to shift from policy-driven privacy architecture to protocol-driven privacy architecture. Therefore, we have introduced a new protocol-driven (i.e., scheme level privacy support) perfect privacy-preserving information retrieval scheme using a concept called private information retrieval (PIR).

Private information retrieval [7] is one of the ways of reading the bit information from the other party privately, and private block retrieval (PBR), a realistic extension of PIR, is the way of reading single block information from the database privately. We have successfully constructed a new PBR scheme in a single database setting which neither belongs to information-theoretic (i.e., no replicated database) nor belongs to computationally bounded (i.e., no privacy assumptions). The proposed scheme fully supports "perfect privacy," i.e., all the queries are mutually exclusive and give no information (not even partial information) about the user privacy. Note that the proposed scheme conventionally uses the term "PIR" but PBR by default until and unless externally stated. Note that the *privacy* refers to the user privacy, and *perfect privacy* refers to zero percent privacy leak until and unless externally stated.

The construction uses "quadratic residuosity" as the underlying data privacy operation. Note that the quadratic residuosity property is only used for preserving data from the intermediate adversaries. This property is not related to hide the privacy of the user, i.e., even on identifying the quadratic residuosity property of the numbers sent in the query, server gains no information about the user's interest. By this, we claim that the construction supports perfect privacy, and the success probability of identifying user's interest is equal to "random guessing."

Finally, we have achieved the following results.

- We have successfully overcome the trivial database download requirement claimed by [7] by achieving the overall communication cost as $o(n)$ where n is the database size which is surely a non-trivial communication as claimed by [9].
- The protocol is generic in nature and can be adopted by both client-server and peer-to-peer privacy critical applications.

Related work: Extensive work has been carried out on PIR by various researchers to fulfill the trade-off between communication and computation overheads, to preserve the user as well as server privacy, handling fault tolerance and integrity. The PIR is mainly classified into two categories in which one relay on non-colluding server replication and the other relay on single database with limited computation power.

Information-Theoretic PIR (itPIR) In order to provide protocol-driven privacy, Chor et al. [7] introduced the concept of private reading from k replicated databases and further improved the communication cost in [8] using XOR operations. There are several other improvements introduced by [1, 2] over communication and computation overheads in itPIR setting. Gertner et al. [13] highlight on the data privacy of the server along with the user privacy using the concept of conditional disclosure of secrets.

Computationally Bounded PIR (cPIR) The first quadratic residuosity assumption-based privacy-preserving PIR scheme was introduced by [16] in a single database setting with sub-polynomial communication cost. Chor and Gilboa [6] also presented a one-way function-based PIR scheme with the minimal database replication to achieve only computationally bounded privacy. Cachin et al. [4] presented ϕ-hiding-based scheme with polylogarithmic communication cost. Ishai et al. [15] introduced an efficient cPIR scheme using anonymity techniques. Aguilar-Melchor and Gaborit [19] introduced fast cPIR scheme based on coding theory and lattice assumptions. Groth et al. [14] proposed multi query cPIR with constant communication rate. Jonathan and Andy [24] improve computational complexity of existing cPIR using trapdoor groups. Kushilevitz and Ostrovsky [17] presented a computationally intractable cPIR using one-way trapdoor permutations. Chang [5] presented a computationally bounded PIR with logarithmic communication using Paillier cryptosystem as the underlying intractability assumption. Gentry and Ramzan [11] presented a PBR scheme with log-squared communication using a decision subgroup problem called ϕ-hiding assumption. In order to protect both user and server privacy, several *oblivious transfer* (OT) schemes [9, 18, 20, 21] have also been introduced in a single database setting. The first keyword-based PIR search [3] has been introduced to apply PIR on the existing server data structure.

Perfect Privacy The term "perfect privacy" as defined in [7] strongly suggests the requirement of the uniformly distributed probability for any two random variables (PIR queries are treated as the random variables). The first information-theoretic single-database PIR scheme was introduced by [12] and recently by [22]. In order to preserve the user privacy in multiuser setting, input anonymity by secret sharing technique is presented by Toledo et al. [23].

Organization: The rest of the paper is organized as follows. The required notations and preliminaries are described in Sect. 6.2; the preliminary modules, the proposed PIR scheme, and the performance analysis along with the required security proofs are all described in Sect. 6.3; and, finally, the open problems are listed along with the conclusion in Sect. 6.4.

6.2 Notations and Preliminaries

Let $[u]$=$\{1, 2, \cdots, u\}$ and $[1, u]$ be the method of selecting all the integers from 1 to u; $\mathcal{DB}_u^{b_v}$ is a set of u number of v bit matrix. Let N=pq (where $N \overset{R}{\leftarrow} \{0, 1\}^k$ with the security parameter k) be the RSA composite modulus, and $\mathcal{S}_{QR}$, $\mathcal{S}_{QNR} \subseteq \mathbb{Z}_N^{+1}$ are the quadratic residue and non-residue subsets respectively. Let $\mathcal{JS}$ and $\mathcal{LS}$ be the *Jacobi* and *Legendre* symbols respectively. Let c be the total number of l-bit groups of a database block. Let $\mathcal{U}$ be the end user or the client or the intended service seeker and $\mathcal{S}$ be the server or the intended service provider.

Quadratic residuosity: $\forall x, y \in \mathbb{Z}_N^{+1}$, if $x \equiv y^2 \pmod N$ then $x \in \mathbb{Z}_N^{+1} \backslash \mathcal{S}_{QNR}$, i.e., $x \in \mathcal{QR}$; otherwise $x \in \mathbb{Z}_N^{+1} \backslash \mathcal{S}_{QR}$, i.e., $x \in \mathcal{S}_{QNR}$.

Definition 1 (Trapdoor Function of [10]) : $\forall x \in \mathbb{Z}_N^*$, $r \in \mathbb{Z}_N^{-1}$, $s \in \mathcal{S}_{QNR}$, $\forall jx, hx \in \{0, 1\}$, the function $\mathcal{T}(x, r, s) = (x)^2 \cdot r^{jx} \cdot s^{hx}$=$z$ such that jx=1 if $\mathcal{JS}_N(x)$=-1 otherwise $jx = 0$ and hx=1 if $x > \frac{N}{2}$ otherwise hx=0. The inverse function $\mathcal{T}^{-1}$ is defined as $\mathcal{T}^{-1}(z) = \sqrt{(z) \cdot r^{-jx} \cdot s^{-hx}}$=$x$. The generalized formula for l number of inputs is given as $\mathcal{T} : (x_1, \cdots, x_l, r, s) \rightarrow z_1, \cdots, z_l$ where $\mathcal{T}(x_1, \cdots, x_l, r, s)$=$\mathcal{T}_1, \cdots, \mathcal{T}_l$ and $\mathcal{T}_i(x_i, r, s)$=$(x_i)^2 \cdot r^{jx_i} \cdot s^{hx_i}$=$z_i$, $i \in [1, l]$.

We have used a slightly modified version of the above trapdoor function for our proposed PIR scheme and is given as $\mathcal{MT} : x \rightarrow (z, t), t \in \{0, 1\}$ where $\mathcal{MT}$=x^2=z and t is assigned with the "hx" value of the input x. The generalized formula for l number of inputs is given as $\mathcal{MT} : (x_1, \cdots, x_l) \rightarrow ((z_1, \cdots, z_l), (t_1, \cdots, t_l))$ where $\mathcal{MT}(x_1, \cdots, x_l)$=$\mathcal{MT}_1, \cdots, \mathcal{MT}_l$ and $\mathcal{MT}_i(x_i)$=x_i^2=(z_i,t_i), $i \in [1, l]$.

Definition 2 (Perfect Privacy PIR Query) : If any two randomly selected PIR queries are independent of block reference (or block index), i.e., $Pr[Q_i \overset{R}{\leftarrow} \mathcal{QF}(1^k) : A(n, Q_i, 1^k) = 1]$ is equal to $Pr[Q_j \overset{R}{\leftarrow} \mathcal{QF}(1^k) : A(n, Q_j, 1^k) = 1]$ where A is a distinguishing server, $\mathcal{QF}$ is the query generating function, and Pr is the probability distribution function then the mutual information between them is $I(Q_i, Q_j)$=0. This implies that the queries are independent of privacy or the PIR queries are exhibiting perfect privacy.

Definition 3 (Perfect Privacy Single Database PIR (perfectPIR)) It is a 5-tuple ($\mathcal{U}$, $\mathcal{S}$, $\mathcal{QF}$, $\mathcal{RC}$, $\mathcal{IE}$) protocol where $\mathcal{U}$ is the customer, $\mathcal{S}$ is the service provider, $\mathcal{QF}$ is the query formulation algorithm run by $\mathcal{U}$, $\mathcal{RC}$ is the response creation algorithm run by $\mathcal{S}$, and $\mathcal{IE}$ is the interest extraction algorithm run by $\mathcal{U}$. Let n bit database $\mathcal{DB}_u^{b_v}$ be a two-dimensional matrix of u rows and v columns. For any interested block $\mathcal{DB}_i$, $i \in [u]$, of the database $\mathcal{DB}_u^{b_v}$, the user $\mathcal{U}$ generates PIR query described in Definition 2 to achieve the user privacy and sends to the database server $\mathcal{S}$ where all the queries sent over the insecure communication channel are coved under "quadratic residuosity assumption" (QRA) to achieve "data privacy." The database server $\mathcal{S}$ replies by generating block-specific response ciphertext set R_j and trapdoor bit set as communication bits for all the blocks $\mathcal{DB}_j$, $j \in [1, u]$ where all the generated ciphertexts from $\mathcal{S}$ are coved under QRA to achieve "data privacy." In turn, user $\mathcal{U}$ retrieves or reads required block $\mathcal{DB}_i$ using the block-specific response ciphertext set R_i and its corresponding trapdoor bit set.

6.3 Perfect Privacy PIR Scheme

Let the database $\mathcal{DB}_u^{b_v}$ be viewed as a two-dimensional matrix of u rows and v columns where n=uv. The database $\mathcal{DB}_u^{b_v}$ of size n=uv is constituted by individual matrix or a block $\mathcal{DB}_i$=$b_1, b_2, \cdots, b_v$,$i \in [u]$, each of size v where b is the bit of $\mathcal{DB}_i$. Let us consider sufficiently large RSA composite modulus N=pq where $p \equiv q \equiv 3$ (mod 4). Assume that both the parties (user and server) have exchanged some prior information like the database size n, c, and l where each database block is divided into c number of l-bit groups and public key combination table (as described in Table 6.1).

At the high level design, the proposed 5-tuple protocol of Definition 3 is viewed as a way of retrieving or reading information from the service provider privately. The intended service seeker or the customer $\mathcal{U}$ wishes to retrieve some information from the intended service provider $\mathcal{S}$ privately using user-centric "public key

Table 6.1 Public key combinations for 2-Bit and 3-bit encoding where Q∈ $\mathbb{Z}_N^{+1} \setminus \mathcal{S}_{QNR}$, N∈ $\mathbb{Z}_N^{+1} \setminus \mathcal{S}_{QR}$

	f_1f_2	f_1f_2	f_1f_2	f_1f_2	$f_1f_2f_3$	$f_1f_2f_3$	$f_1f_2f_3$	$f_1f_2f_3$	$f_1f_2f_3$	$f_1f_2f_3$	$f_1f_2f_3$	$f_1f_2f_3$
Input	0 0	0 1	1 0	1 1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Public key combinations	2 2	2 1	1 2	1 1	2 2 2	2 2 1	2 1 2	2 1 1	1 2 2	1 2 1	1 1 2	1 1 1
Output property	QQ	QN	NQ	NN	QQQ	QQN	QNQ	QNN	NQQ	NQN	NNQ	NNN

Conventionally, in the second row, 2 ⇔ $\mathcal{PK}_2$ and 1 ⇔ $\mathcal{PK}_1$

cryptography." In order to achieve private retrieval or private reading to read the block from the server, $\mathcal{U}$ generates perfect privacy supported PIR query $\mathcal{Q}$ (i.e., query is selected as described in Definition 2) using the *initialization* or query formulation algorithm $\mathcal{QF}$ and sends to $\mathcal{S}$. The service provider $\mathcal{S}$ generates and sends back the response by involving all the database blocks using *reply* or response creation algorithm $\mathcal{RC}$. Finally, the customer $\mathcal{U}$ retrieves the required block privately using *reading* or interest extraction algorithm $\mathcal{IE}$.

***l*-Bit Input v/s *l*-Output Property Combination:** The public key combination selection for a particular bit or a group of bits is described as follows. Let us consider an encoding function $f : (b \overset{R}{\leftarrow} \{0,1\}, x \overset{R}{\leftarrow} \mathbb{Z}_N^*, \mathcal{PK} \overset{R}{\leftarrow} \mathbb{Z}_N^{+1}) \rightarrow (z \in \mathbb{Z}_N^{+1})$ using the encoding bit b, random input x, and public key $\mathcal{PK}$ as

$$f(b, x, \mathcal{PK}) = \begin{cases} (x^2 \cdot \mathcal{PK} \mid \mathcal{PK} \overset{R}{\leftarrow} \mathcal{S}_{QNR}) \equiv (z \in \mathcal{S}_{QNR}) \pmod{N} & \text{ifb} = 1 \\ (x^2 \cdot \mathcal{PK} \mid \mathcal{PK} \overset{R}{\leftarrow} \mathcal{S}_{QR}) \equiv (z \in \mathcal{S}_{QR}) \pmod{N} & \text{ifb} = 0 \end{cases} \tag{6.1}$$

If the encoding bit b=1, then the public key $\mathcal{PK}$ should always be selected from $\mathcal{S}_{QNR}$ so that the output ciphertext z always resides in $\mathcal{S}_{QNR}$. Similarly, if the encoding bit b=0, then the public key $\mathcal{PK}$ should always be selected from $\mathcal{S}_{QR}$ so that the output ciphertext z always resides in $\mathcal{S}_{QR}$. If there are l-bit input functions $f_1, \cdots, f_l$ producing l output ciphertexts and each function drawn from (6.1) encodes one bit, then l public key combinations are to be used to encode l-bit input. For instance, for 2-bit input, there are two encoding functions f_1,f_2, two public keys $\mathcal{PK}_1 \overset{R}{\leftarrow} \mathcal{S}_{QNR}$,$\mathcal{PK}_2 \overset{R}{\leftarrow} \mathcal{S}_{QR}$, and four public key combinations, namely, (($\mathcal{PK}_1$,$\mathcal{PK}_1$),($\mathcal{PK}_1$,$\mathcal{PK}_2$),($\mathcal{PK}_2$,$\mathcal{PK}_1$),($\mathcal{PK}_2$,$\mathcal{PK}_2$)) as shown in Table 6.1. Similarly, for 3-bit input, there are three encoding functions f_1,f_2,f_2, two public keys $\mathcal{PK}_1$,$\mathcal{PK}_2$, and eight public key combinations, namely, (($\mathcal{PK}_1$,$\mathcal{PK}_1$,$\mathcal{PK}_1$),($\mathcal{PK}_1$, $\mathcal{PK}_1$, $\mathcal{PK}_2$),($\mathcal{PK}_1$,$\mathcal{PK}_2$, $\mathcal{PK}_1$),($\mathcal{PK}_1$,$\mathcal{PK}_2$,$\mathcal{PK}_2$), ($\mathcal{PK}_2$, $\mathcal{PK}_1$,$\mathcal{PK}_1$), ($\mathcal{PK}_2$,$\mathcal{PK}_1$,$\mathcal{PK}_2$), ($\mathcal{PK}_2$,$\mathcal{PK}_2$,$\mathcal{PK}_1$),($\mathcal{PK}_2$, $\mathcal{PK}_2$, and $\mathcal{PK}_2$)) as shown in Table 6.1. If the input bit is 0, then always $\mathcal{PK}_1$ is selected; otherwise $\mathcal{PK}_2$ is selected in the encoding function. Clearly, in order to get unique l ciphertext output quadratic residuosity property combinations (as shown in Table 6.1), public key $\mathcal{PK}_1$ is selected if the input bit is

1 otherwise public key $\mathcal{PK}_2$ is selected during encoding process using the encoding function f.

Bit Group Encoding: Let us view the database block $\mathcal{DB}_i$=$b_1, b_2, \cdots, b_v$,$i \in [u]$ as a set of l-bit groups $\{G_1 = (b_1, \cdots, b_l), G_2 = (b_{l+1}, \cdots, b_{2l}), G_3 = (b_{2l+1}, \cdots, b_{3l}), \cdots, G_\sigma = (b_{v-l+1}, \cdots, b_v)\}$ for some v=lc where $c > 0$ is an integer constant and $\sigma \in [1, c]$. In order to accomplish PIR operation on a block $\mathcal{DB}_i$, the PIR encoding function for $l \in \mathbb{N}$ bit group G_σ, $(y_1, \cdots, y_l) \overset{R}{\leftarrow} \mathbb{Z}_N^*$ bit input and public keys $\mathcal{PK}_1 \overset{R}{\leftarrow} \mathcal{S}_{QNR}$,$\mathcal{PK}_2 \overset{R}{\leftarrow} \mathcal{S}_{QR}$ is $\mathcal{E} : (G_\sigma, N, y_1, \cdots, y_l, \mathcal{PK}_1, \mathcal{PK}_2) \rightarrow (\alpha_1, \cdots, \alpha_l)$ where $\alpha_1, \cdots, \alpha_l$ are the corresponding ciphertext outputs. The detailed description of the encoding function $\mathcal{E}$ is as follows.

$$\mathcal{E}_\sigma(G_\sigma, N, y_1, \cdots, y_l, \mathcal{PK}_1, \mathcal{PK}_2) = \begin{cases} f_1 = [(y_1)^2 \cdot \mathcal{PK}_j \equiv \alpha_1 \pmod{N}] \\ \cdot = [\ \cdot \qquad \cdot \ \equiv \ \cdot \qquad\qquad] \\ \cdot = [\ \cdot \qquad \cdot \ \equiv \ \cdot \qquad\qquad] \\ f_l = [(y_l)^2 \cdot \mathcal{PK}_{j'} \equiv \alpha_l \pmod{N}] \end{cases} \tag{6.2}$$

where $j, j^{'} \in [2]$ and each f which encodes one bit of G_σ in (6.2) is drawn from (6.1).

Connecting Two Encoding Functions Using [10]: For any two consecutive PIR encoding functions $\mathcal{E}_\sigma$ and $\mathcal{E}_{\sigma+1}$ of (6.2) where $1 \leq \sigma \leq (c-1)$, the connecting function C is described as follows. Let us consider $\mathcal{E}_\sigma : (G_\sigma, N, y_1, \cdots, y_l, \mathcal{PK}_1, \mathcal{PK}_2) \rightarrow \{\alpha_1, \alpha_2, \cdots, \alpha_l\}$ and $\mathcal{E}_{\sigma+1}$:$(G_{\sigma+1}, N, \alpha_1, \cdots, \alpha_l, \mathcal{PK}_1, \mathcal{PK}_2) \rightarrow \{\alpha_1^{'}, \alpha_2^{'}, \cdots, \alpha_l^{'}\}$ then the connecting function $C : (\mathcal{E}_\sigma, \mathcal{E}_{\sigma+1}) \rightarrow (\{\alpha_1^{'}, \alpha_2^{'}, \cdots, \alpha_l^{'}\}, \{t_1, t_2, \cdots, t_l\})$ where each *trapdoor bit* t_i, $i \in [l]$ generated from the modified trapdoor function $\mathcal{MT}$ of (6.3) and is equivalent to "hx" value of the trapdoor function $\mathcal{T}$ described in Definition 1. Each connecting function C in turn connects to the next connecting function.

$$C(\mathcal{E}_\sigma, \mathcal{E}_{\sigma+1}) = \begin{cases} \qquad\qquad \text{Apply } \mathcal{E}_\sigma \text{ first} \\ \qquad\qquad\qquad \text{then} \\ \mathcal{E}_{\sigma+1} = \begin{cases} (\mathcal{MT}(\alpha_1))^2 \cdot \mathcal{PK}_j \equiv \alpha_1^{'} \pmod{N} \\ \quad \cdot \qquad\quad \cdot \ \equiv \ \cdot \\ \quad \cdot \qquad\quad \cdot \ \equiv \ \cdot \\ (\mathcal{MT}(\alpha_l))^2 \cdot \mathcal{PK}_{j'} \equiv \alpha_l^{'} \pmod{N} \end{cases} \end{cases} \tag{6.3}$$

Note that only $\mathcal{E}_1$ selects the input $y_1, \cdots, y_l$ from $\mathbb{Z}_N^*$, and all other $\mathcal{E}_i$, $i \in [2, c]$, select $\alpha_1, \cdots, \alpha_l$ from $\mathbb{Z}_N^{+1}$.

6.3.1 Generic l-Bit Perfect Privacy PIR Scheme

By combining above modules, we have finally constructed a perfect privacy preserving single database PIR as follows. All the below described PIR algorithms are taken from Definition 3.

- **Initializing** ($\mathcal{QF}$): User $\mathcal{U}$ sends a block independent single PIR query Q=(N, $\mathcal{PK}_1$, $\mathcal{PK}_2$, y_1,$\cdot\cdot\cdot$, y_l) to the server where $\mathcal{PK}_1 \overset{R}{\leftarrow} \mathcal{S}_{QR}$, $\mathcal{PK}_2 \overset{R}{\leftarrow} \mathcal{S}_{QNR}$ and $y_1,\cdot\cdot\cdot, y_l \overset{R}{\leftarrow} \mathbb{Z}_N^*$.
- **Reply** ($\mathcal{RC}$): Server generates the block-specific response R_i, $i \in [1, u]$, as follows. As a result of response, each block $\mathcal{DB}_i$ generates two ciphertexts and trapdoor bit set as

$$\begin{aligned} cl\text{Block PIR encryption} &= \mathcal{E}_i(G_i, N, \mathcal{MT}(\mathcal{E}_{i-1}), \mathcal{PK}_1, \mathcal{PK}_2) \\ &= ((\beta_i^{\alpha_l} = (\alpha_1, \cdot\cdot\cdot, \alpha_l)), (\rho_i^{t_{l(c-1)}} = (t_1, \cdot\cdot\cdot, t_{l(c-1)}))) \\ &= R_i \end{aligned} \tag{6.4}$$

where $i \in [c, 2]$, $\beta_i^{\alpha_l}$ is l number of ciphertexts generated at the block i, $\rho_i^{t_{l(c-1)}}$ is $l(c-1)$ number of trapdoor bits generated at the block i, $\mathcal{E}_1(\mathcal{PK}_j, \mathcal{PK}_{j'}, y_1, \cdot\cdot\cdot, y_l, G_1)$. Note that any two consecutive PIR encoding functions $\mathcal{E}_\sigma$ and $\mathcal{E}_{\sigma-1}$, $c \geq \sigma \geq 2$ described as $\mathcal{E}_\sigma(G_\sigma, N, \mathcal{MT}(\mathcal{E}_{\sigma-1}), \mathcal{PK}_1, \mathcal{PK}_2)$ in (6.4) is equivalent to the connecting function $C(\mathcal{E}_{\sigma'}, \mathcal{E}_{\sigma'+1})$, $1 \leq \sigma' \leq (c-1)$. The overall response from all the blocks of $\mathcal{DB}_u^{b_v}$ would be R=$R_1||R_2|| \cdot\cdot\cdot ||R_u$. The response R is sent back to the user.
- **Reading** ($\mathcal{IE}$): By using the block-specific response R_i=($\beta_i^{\alpha_l}$, $\rho_i^{t_{l(c-1)}}$), the user privately reads the required bits of the interested block $\mathcal{DB}_i$ as follows.

$$\mathcal{D}(p, q, \mathcal{E}_i(\mathcal{PK}_1, \mathcal{PK}_2, \mathcal{MT}^{-1}(\mathcal{D}(p, q, \mathcal{E}_{i+1})))=(b_1, b_2, \cdot\cdot\cdot, b_v)=\mathcal{DB}_i$$

or

$$\mathcal{D}(p, q, \beta_i^{\alpha_l}, \rho_i^{t_{l(c-1)}}) = (b_1, b_2, \cdot\cdot\cdot, b_v) = \mathcal{DB}_i \tag{6.5}$$

where $i \in [1, c-1]$, $\mathcal{E}_c(\mathcal{PK}_j, \mathcal{PK}_{j'}, \alpha_1, \cdot\cdot\cdot, \alpha_l, t_{l(c-1)})$.

Theorem 1 *If any two randomly selected PIR queries are independent to each other, then they exhibit perfect privacy in PIR environment. In other words, for all quadratic residuosity-based perfect privacy PIR protocols, the probability distributions of any two randomly selected queries are always equal and independent to each other, and hence mutual information between those two queries is always zero.*

Proof (Sketch) Consider any PIR query Q=(N, $\mathcal{PK}_1$, $\mathcal{PK}_2$, y_1,$\cdot\cdot\cdot$, y_l) constructed by the user in $\mathcal{QF}$ algorithm. Note that the domains of each element are $\mathcal{PK}_1 \overset{R}{\leftarrow} \mathcal{S}_{QR}$, $\mathcal{PK}_2 \overset{R}{\leftarrow} \mathcal{S}_{QNR}$ and $y_1,\cdot\cdot\cdot, y_l \overset{R}{\leftarrow} \mathbb{Z}_N^*$. Also consider $y_1',\cdot\cdot\cdot, y_l' \overset{R}{\leftarrow} \mathbb{Z}_N^*$. Since the domain of the query input is always $\mathbb{Z}_N^*$ or the query input is always independent of the block index, it is intuitive that $Pr[Q_i = (N, \mathcal{PK}_1, \mathcal{PK}_2, y_1, \cdots, y_l) \xleftarrow{R} \mathcal{QF}(1^k) : A(n, Q_i, 1^k) = 1]$ is equal to $Pr[Q_j = (N, \mathcal{PK}_1, \mathcal{PK}_2, y_1^{'}, \cdots, y_l^{'}) \xleftarrow{R} \mathcal{QF}(1^k) : A(n, Q_i, 1^k) = 1]$. Therefore, the randomly selected queries $X = Q_i$ and $Y = Q_j$ or random variables X and $Y$are independent to each other. Intuitively, $Pr(XY)$= $Pr(X, Y)$= $Pr(X) \cdot Pr(Y)$ provided $Pr(Y)$ >0. The respective conditional distribution of X and Y and the mutual informations are

$$\boldsymbol{Pr(X \mid Y)} = \frac{Pr(XY)}{Pr(Y)} = \frac{Pr(X) \cdot Pr(Y)}{Pr(Y)} = Pr(X) \tag{6.6}$$

$$\begin{aligned}\boldsymbol{I(X, Y)} = \textstyle\sum_X \sum_Y Pr(X, Y)\, log\, \frac{Pr(X) \cdot Pr(Y)}{Pr(X) \cdot Pr(Y)} &= \textstyle\sum_X \sum_Y Pr(X, Y)\, log\, 1 \\ &= \textstyle\sum_X \sum_Y Pr(X, Y) \cdot 0 = 0\end{aligned} \tag{6.7}$$

Intuitively, all the PIR queries are mutually exclusive. This implies user privacy is independent of its query input. Therefore, all the PIR queries exhibit perfect privacy, i.e., the server gains no knowledge about the user privacy or block that the user wishes to retrieve by the query analysis using his unlimited computing power.

Theorem 2 *For all perfectPIR protocol, there exists a communication cost $o(n)$ which is always less than the trivial database download cost $O(n)$.*

Proof (Sketch) After each PIR encoding function $\mathcal{E}_i$, $i \in [1, c]$, the modified trapdoor function $\mathcal{MT}$ generates l number of trapdoor bits. Since there are c number of l bit groups present or equivalently lc number of intermediate ciphertexts generated in a block $\mathcal{DB}_j$, $j \in [u]$, there are exactly $l(c - 1)$ or $(v - l)$ number of trapdoor bits generated from each block. In total, there are $ul(c - 1)$ or $u(v - l)$ number of trapdoor bits generated from the entire database. Clearly, $ul(c - 1)$ or $u(v - l)$ is always less than the database size uv. Therefore the communication cost w.r.t the database size is always $o(n)$ which is clearly an acceptable communication cost for "perfect privacy" in PIR environment.

Performance: User generates $k(3 + l)$ bit length PIR query Q and sends it to the server. Server generates $l(k + (c - 1))$ number of communication bits from each block and hence $u[l(k + (c - 1))]$ number of bits from the entire database and sends back this communication bits to the user. Server performs $2ulc$ number of modular multiplications during PIR invocation, and user performs only $2lc$ number of modular multiplications during block retrieval.

6.4 Conclusion with Open Problems

We have successfully constructed a new perfect privacy-preserving information retrieval protocol with $o(n)$ communication cost. The proposed scheme successfully adopts quadratic residuosity-based public key cryptography as the underlying primitive. In the future, it is essential to reduce the communication cost so that all including the bandwidth-limited applications can adopt the scheme. Therefore, constructing a perfect privacy PIR with the efficient communication cost is still an open problem.

References

1. Beimel, A., Ishai, Y.: Information-theoretic private information retrieval: a unified construction. In: Proceedings of 28th ICALP, pp. 912–926. Springer, Berlin (2001)
2. Beimel, A., Stahl, Y.: Robust information-theoretic private information retrieval. J. Cryptol. **20**(3), 295–321 (2007)
3. Benny, C., Niv, G., Moni, N.: Private information retrieval by keywords. Cryptology ePrint Archive, Report 1998/003 (1998). http://eprint.iacr.org/1998/003
4. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Proceeedings of 17th Theory and Application of Cryptographic Techniques, EUROCRYPT'99, pp. 402–414. Springer, Berlin (1999)
5. Chang, Y.C.: Single Database Private Information Retrieval with Logarithmic Communication, pp. 50–61. Springer, Berlin (2004)
6. Chor, B., Gilboa, N.: Computationally private information retrieval (extended abstract). In: Proceedings of 29th STOC, pp. 304–313. ACM, New York (1997)
7. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Proceedings of the 36th FOCS, pp. 41–50. IEEE Computer Society, Washington (1995)
8. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. J. ACM **45**(6), 965–981 (1998)
9. Di Crescenzo, G., Malkin, T., Ostrovsky, R.: Single Database Private Information Retrieval Implies Oblivious Transfer, pp. 122–138. Springer, Berlin (2000)
10. Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. Cryptology ePrint Archive, Report 2009/590 (2009). http://eprint.iacr.org/2009/590
11. Gentry, C., Ramzan, Z.: Single-database private information retrieval with constant communication rate. In: Proceedings of 32nd ICALP, pp. 803–815. Springer, Berlin (02005)
12. Gertner, Y., Goldwasser, S., Malkin, T.: A random server model for private information retrieval or how to achieve information theoretic pir avoiding database replication. In: Proceedings of 2nd RANDOM, pp. 200–217. Springer, Berlin (1998)
13. Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. In: Proceedings of 13th STOC, pp. 151–160. ACM, New York (1998)
14. Groth, J., Kiayias, A., Lipmaa, H.: Multi-query computationally-private information retrieval with constant communication rate. In: Proceedings of 13th PKC, pp. 107–123. Springer, Berlin (2010)
15. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Cryptography from anonymity. In: Proceedings of 47th FOCS, pp. 239–248. IEEE Computer Society, Washington (2006)
16. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: Proceedings of 38th FOCS, pp. 364–. IEEE Computer Society, Washington (1997)
17. Kushilevitz, E., Ostrovsky, R.: 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, pp. 104–121. Springer, Berlin (2000)
18. Lipmaa, H.: An oblivious transfer protocol with log-squared communication. In: Proceedings of 8th ISC, pp. 314–328. Springer, Berlin (2005)

19. Melchor, C.A., Gaborit, P.: A lattice-based computationally-efficient private information retrieval protocol (2007)
20. Naor, M., Pinkas, B.: Oblivious transfer and polynomial evaluation. In: Proceedings of 31st STOC, pp. 245–254. ACM, New York (1999)
21. Rabin, M.O.: How to exchange secrets with oblivious transfer. Harvard University Technical Report (2005)
22. Shah, N.B., Rashmi, K.V., Ramchandran, K.: One extra bit of download ensures perfectly private information retrieval. In: IEEE International Symposium on Information Theory, pp. 856–860 (2014)
23. Toledo, R.R., Danezis, G., Goldberg, I.: Lower-cost epsilon-private information retrieval. CoRR abs/1604.00223 (2016)
24. Trostle, J., Parrish, A.: Efficient computationally private information retrieval from anonymity or trapdoor groups. In: Proceedings of 13th ISC, pp. 114–128. Springer, Berlin (2011)