T-79.511 Special Course on Cryptology / Private Information Retrieval

Vesa Vaskelainen

30.10.2003

Abstract

This survey is predominantly based on the papers written by Gertner et al. [2] and Kerenidis and Wolf [4]. Private information retrieval schemes are examined from classical and quantum information theoretic point of view.

1 Introduction

Private Information Retrieval (PIR) schemes allow a user to retrieve information from a database while maintaining his query private. We view the database as a binary string $x = x_1 \dots x_n$ of length n. Identical copies of this string are stored by $k \geq 2$ servers. The user has some index i, and he is interested in obtaining the value of the bit x_i. To achieve this goal, the user queries each of the servers and gets replies from which the desired bit x_i can be computed. The query to each server is distributed independently of i and therefore each server gains no information about i. [1]

Symmetrically Private Information Retrieval (SPIR) guarantees also the privacy of the data, as well as of the the user. This means that, in addition to maintaining the privacy of the user, every invocation of SPIR, in addition to maintaining the user privacy, prevents the user (even a dishonest one) from obtaining any information other than a single physical bit of the data. Data privacy is a natural and crucial requirement in many settings. For example, consider a commercial database which sells information, such as stock information, to users, charging by the amount of data that the user retrieved. Here, both user privacy and database privacy are essential. [2]

In the end we will study the existence of SPIR schemes in the quantum world, where user and servers have quantum computers and can communicate qubits. The setting of SPIR was introduced by Gertner et al. [2]. They showed that honest user SPIRs without shared randomness are impossible in the classical world. The necessity of shared randomness for classical SPIR schemes is a significant drawback, since information-theoretic security requires new shared randomness for each application of the scheme. This either requires a lot of extra communication between the servers (if new shared randomness is generated for each new application) or much memory on the parts of the server (if randomness is generated once for many applications, each server needs to store this). [4]

2 Notation and Definitions

The following notations and conventions are used throughout this survey. By $[l]$ is denoted the set $1, 2, \dots, l$. For any sets $S, S' \subseteq [l]$, we let $S \oplus S'$ denote the symmetric difference between S and S' (i.e., $S \oplus S' = (S \backslash S') \cup (S' \backslash S)$), and χ_S denotes the characteristic vector of S: an l-bit binary string whose j-th bit is equal to 1 iff $j \in S$. To simplify notation, $S \oplus j$ and χ_j are used instead of $S \oplus \{j\}$ and $\chi_{\{j\}}$, respectively. $\{0,1\}^n$ means the set of strings of length n with each letter being either zero or one.[2]

By default, the terms "PIR scheme" and "SPIR scheme" refer to 1-round, information theoretically private schemes which means that no computational assumptions are made. Complexity is measured, by default, in terms of communication. PIR scheme must satisfy the *user privacy* requirement: under any two indices i, i', the communication seen by any single database is indentically distributed.

The *data privacy* condition of SPIR schemes requires that for any user (possibly a dishonest one, not following the protocol) interacting with the honest databases $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$ there exists an index i s.t. for every data strings x, x' satisfying $x_i = x'_i$ the distribution of communication is independent of the data strings x and x'. An honest-user-SPIR scheme is a PIR scheme that satisfies the data-privacy requirement with respect to an honest (but curious) user, who follows the specification of the scheme but may try to deduce extra information from the communication. [2]

2.1 Very Short Introduction to Quantum Mechanics

A good understanding of quantum mechanics is based on a good knowledge of elementary linear algebra which is the study of vector spaces and of operations on those vector spaces. The basic objects of linear algebra are vector spaces. For us the vector space of most interest is $\mathbb{C}^n$, the n-dimensional complex vector space. Vectors are the elements of a vector space. The standard quantum mechanical notation for a vector in a vector space is,

$$|\psi\rangle$$

where ψ is a label for the vector, mostly labels ψ and ϕ are used. The $|\cdot\rangle$ notation indicates that the object is a column vector. The entire object $|\psi\rangle$ is also known as *ket* and its vector dual (a row vector) $\langle\psi|$ is known as *bra*.

The most fundamental entity in information science is the *bit* which is a system which carries either "0" or "1" value. The quantum analog of a bit is *qubit* which is two- state system where the two possible states are called $|0\rangle$ and $|1\rangle$. Basically any quantum mechanical system which has at least two states can be a qubit. When quantum states are used to encode bits the most essential property of them is the possibility of coherence and superposition. The general state is,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$. This means that qubit is in a superposition of both states and when the qubit is measured we will find it with probability $|\alpha|^2$ to carry value "0" and with probability $|\beta|^2$ to carry value "1". It is sufficient to think of qubits in abstract terms, without reference to a specific realization.

The *tensor product* is a way of putting vector spaces together to from larger vector spaces. Suppose V and W are vector spaces of dimension m and n respectively. For convenience we also suppose that V and W are Hilbert spaces. Then $V \otimes W$ (read 'V tensor W') is an mn dimensional vector space. The elements of $V \otimes W$ are linear combinations of 'tensor products' $|v\rangle \otimes |w\rangle$ of elements $|v\rangle$ of V and $|w\rangle$ of W. Often is used abbreviated notation $|v\rangle|w\rangle$, $|v, w\rangle$ or even $|vw\rangle$ for the tensor product $|v\rangle \times |w\rangle$. [5]

3 The PIR schemes

Here is reviewed an elementary PIR scheme introduced by Chor et al. denoted by *basic cube scheme* [1]. This scheme will be the basis for the other shemes described later.

3.1 Basic d-dimensional Cube Scheme

This is a PIR scheme for $k = 2^d$ databases. Assume without loss of generality that the database size is $n = l^d$, where $d, l \in \mathbb{Z}_+$. The index set $[n]$ can then be identified with the d-dimensional cube $[l]^d$, in which each index $i \in [n]$ can be naturally identified with a d-tuple $(i_1, \ldots, i_d)$. Tuple is an ordered sequence and previous can also understood as a coordinate in d-dimensional space. A d-dimensional subcube is a subset $S_1 \times \cdots \times S_d$ of d-dimensional cube, where each S_i is a subset of $[l]$. Such a subcube is represented by the d-tuple $C = (S_1, \ldots, S_d)$. The k databases will be indexed by all binary strings of length d. The scheme proceeds as follows.
QUERIES: The user picks a random subcube $(S_1^0, \ldots, S_d^0)$, where $S_1^0, \ldots, S_d^0$ are independent, random subsets of $[l]$. Let $S_m^1 = S_m^0 \oplus i_m$ $(1 \leq m \leq d)$. For each $\sigma = \sigma_1\sigma_2 \ldots \sigma_d \in \{0,1\}^d$, the user sends to database $\mathcal{DB}_\sigma$ the subcube $C_\sigma = (S_1^{\sigma_1}, \ldots, S_d^{\sigma_d})$, where each set $S_m^{\sigma_m}$ is presented by its characteristic l-bit string.
ANSWERS: Each database $\mathcal{DB}_\sigma, \sigma \in \{0,1\}^d$, computes the exclusive-or of the data bits residing in the subcube C_σ, and sends the resultant bit b_σ to

the user.
RECONSTRUCTION: The user computes x_i as the exclusive-or of the k bits b_σ it has received. [2]

The correctness of the scheme can be proved in the following way. Consider the contribution of each bit $x_{j_1,\ldots,j_d}$ of the database to the sum computed by the user in reconstruction stage. This contribution depends on the number of subcubes (corresponding to the queries directed to the k databases) that contain the position $(j_1,\ldots,j_d)$. It is not hard to see that $(i_1,\ldots,i_d)$ is the only position that is contained in an odd number of subcubes, more specific it appears in a single subcube. This is because, for every $t \in [d]$, the value i_t appears in exactly one of the sets S_t^0, S_t^1. Each of the other positions $(j_1,\ldots,j_d)$ $(\neq (i_1,\ldots,i_d))$ appears in an even number of subcubes: Suppose $j_t \neq i_t$, then for every $\sigma_1,\ldots,\sigma_d$,

$$(j_1,\ldots,j_d) \in S_1^{\sigma_1}\times\cdots\times S_{t-1}^{\sigma_{t-1}}\times S_t^0\times S_{t+1}^{\sigma_{t+1}}\times\cdots\times S_d^{\sigma_d}$$

if and only if

$$(j_1,\ldots,j_d) \in S_1^{\sigma_1}\times\cdots\times S_{t-1}^{\sigma_{t-1}}\times S_t^1\times S_{t+1}^{\sigma_{t+1}}\times\cdots\times S_d^{\sigma_d}$$

Therefore, in the sum modulo 2 computed by the user in reconstruction stage, the contribution of these positions is cancelled and the only value that remains is that of position $(i_1,\ldots,i_d)$. [1]

The communication involved in the above scheme consists of sending a sequence of d subsets in $[l]$ to each server, and receiving a single bit back. Hence the total communication complexity is $k\cdot(d\cdot l+1) = 2^d\cdot(d\cdot\sqrt[d]{n}+1) = \mathcal{O}(n^{1/d})$. [1]

3.2 The PIR Scheme $\mathcal{B}_2$

This scheme may be regarded as a 2-database implementation of the 8-database 3-dimensional cube scheme described above. Let $l = n^{1/3}$, and then let $i = (i_1, i_2, i_3)$ be the index of the data bit being retrieved. Each of two databases $\mathcal{DB}_{000}$ and $\mathcal{DB}_{111}$ emulates the 4 databases $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^3$, such that the Hamming distance of σ from its index is at most 1. This is done in the following way. The user sends to $\mathcal{DB}_{000}$ the subcube $C_{000} = (S_1^0, S_2^0, S_3^0)$ and to $\mathcal{DB}_{111}$ the subcube $C_{111} = (S_1^1, S_2^1, S_3^1)$ as in the basic cube scheme. The database $\mathcal{DB}_{000}$ replies with its own answer bit b_{000} along with 3 l-bit long strings, each of which contains the answer bit of the other databases it emulates. For instance, the i_1'-th bit of the string emulating $\mathcal{DB}_{100}$ is obtained by computing the exlusive-or of all data bits residing in the subcube $(S_1^0 \oplus i_1', S_2^0, S_3^0)$, implying that the i_1-th bit in this string is equal to b_{100}. Symmetrically, $\mathcal{DB}_{111}$ sends the single bit b_{111} along with 3 l-bit long strings, each of which corresponds to the subcubes obtained from C_{111} by "masking" the set S_m^1 with all l possible values of i_m'. The user receives 8 answer strings $a_\sigma, \sigma \in \{0,1\}^3$, six of which contain l bits each, and the other two a_{000} and a_{111} contain single bits. In each of the l-bit long strings, the index of the required answer bit b_σ is either i_1 (for $\sigma = 100, 011$), i_2 ($\sigma = 010, 101$), or i_3 ($\sigma = 001, 110$). Since the user can locate all 8 bits $b_\sigma, \sigma \in \{0,1\}^3$, in the answer strings, it can reconstruct x_i by computing their exclusive-or. [2]

4 The SPIR Schemes

In this section, it will be first shown that the 2-database scheme $\mathcal{B}_2$ can be transformed into an honest-user-SPIR scheme $\mathcal{B}_2'$ of the same asymptotic complexity. That will be used as a basis for a recursive construction of a k-database honest-user-SPIR scheme $\mathcal{B}_k'$. In order to do the transformation we need the following tools.

4.1 Conditional Disclosure of Secrets

The model of *conditional disclusure of secrets* consists of: the "condition" - a fixed Boolean function h: $\{0,1\}^n \rightarrow \{0,1\}$ for some n; k players $P_1,\ldots,P_k$; and an external party Carol. Carol holds an input string $y \in \{0,1\}^n$, which is also partitioned between the k players in an arbitrary way (i. e. each player holds some fixed subset of the bits of y). All k players have access to a shared random string, which is hidden from Carol. Finally, a secret input s (single bit unless otherwise mentioned) is known to at least one of the players, but is unknown to Carol. Based on its share of y and on the shared randomness, each player P_j simultaneously sends a message to Carol, s. t. (1) if $h(y) = 1$, then Carol is able to reconstruct the secret s from her input y and from the message she received; and (2) if $h(y) = 0$, then Carol obtains no information (in the information-theoretic sense) about s. [2]

4.2 Private Simultaneous Messages

In this model there are k players, each player P_j holding a private input string y_j, and an external referee called Carol. All players have access to a shared random input, which is unknown to Carol. Based on its private input y_j and the shared random input, each player P_j simultaneously sends a single message to Carol. From the message she received, Carol should be able to compute some predetermined function $f(y_1, \ldots, y_k)$ of the inputs, but should obtain no additional information on the input other than what follows from the value of f.

The PSM complexity of f is the number of communication bits needed to privately compute the funtion f in such a way.Gertner et al. gives a general claim that in principle PSM solution can be applied to any PIR scheme and obtain a honest-user-SPIR scheme, but if reconstruct function has non-linear PSM complexity (even some of the simplest Boolean functions are not known to have linear PSM complexity), this may result in considerable communication overhead. In the next subsection is shown how this problem can be solved for schemes of a certain structure. [2]

4.3 SPIR Schemes Based on Conditional Disclosure of Secrets and PSM

The Theorem 3. from [2] claims that there exsits a 2-database honest-user-SPIR schme, $\mathcal{B}_2'$, of communication complexity $\mathcal{O}(n^{1/3})$. The proof for that is given in the following. The reconstruction function of $\mathcal{B}_2$ may be viewed as a two-stage procedure: (1) the user selects a single bit from each of 8 answer strings, depending only on the index i; and (2) the user exclusive-ors the 8 bits it has selected to obtain x_i. Thus, if we let the honest user learn only the exclusive-or of the 8 bits corresponding to i, the data privacy requirement will be met. This can be achieved by using the conditional disclosure of secrets primitive on top of a PSM protocol computing the exclusive-or of 8 bits. The scheme $\mathcal{B}_2'$, an honest-user-SPIR version of $\mathcal{B}_2$, proceeds as follows:
QUERIES: The user sends the subcube C_{000} to $\mathcal{DB}_{000}$ and C_{111} to $\mathcal{DB}_{111}$, as in the scheme $\mathcal{B}_2$. In addition, the user independently shares the characteristic vectors χ_{i_m}, m = 1,2,3, among the two databases. This is done by picking random l-bit strings r_m^0, r_m^1 such that $r_m^0 \oplus r_m^1 = \chi_{i_m}$ and sending the three strings r_m^0 to $\mathcal{DB}_{000}$ and the three strings r_m^1 to $\mathcal{DB}_{111}$.
ANSWERS: Each of the two databases computes 4 answer strings as in the $\mathcal{B}_2$ scheme. Denote by a_σ the answer string emulating $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^3$. The databases treat each bit of a string a_σ as an input to a PSM protocol computing the XOR of 8 bits, and using their shared randomness they compute the PSM message sent for each such bit. Under the simple PSM protocol for XOR, each such message consists of a single bit. Let w_σ denote the string obtained by replacing each bit from a_σ by its corresponding PSM message bit. In this case, w_σ is obtained by XOR-ing every bit of a_σ with the same random bit r_σ, where the bits $\{r_\sigma\}$ are 8 random bits whose XOR is 0. Finally, for every $\sigma \in \{0,1\}^3$ and $1 \leq j \leq |w_\sigma|$, the database use their shared randomness to disclose to the user the j-th bit of w_σ, $(w_\sigma)_j$, subject to an appropriate condition. For $\sigma = 100, 011$ the condition is $(r_1^0)_j \oplus (r_1^1)_j = 1$, for $\sigma = 010, 101$ it is $(r_2^0)_j \oplus (r_2^1)_j = 1$, and for $\sigma = 001, 110$ it is $(r_3^0)_j \oplus (r_3^1)_j = 1$. The single bits w_{000}, w_{111} can be sent in a plain form.
RECONSTRUCTION: The user reconstructs the eight PSM message bits corresponding to the index i (using the reconstruction function of the conditional disclosure protocol), and computes their exclusive-or to obtain x_i.

The proof of correctness of the $\mathcal{B}_2'$ scheme and user privacy as well as the data privacy requirement are not showed here. Required communication complexity $\mathcal{O}(n^{1/3})$ follows from the fact that each of the $\mathcal{O}(n^{1/3})$ bits of the strings w_j is expressed by a Boolean formula of a constant size and then all such bits can be conditionally disclosed with a total communication cost of $\mathcal{O}(n^{1/3})$. A reader interested in knowing more about proofs is advised to read [3]. However, the above scheme can be generalized to any number of databases $k \geq 2$. (see proof from [2], Appendix B)

Theorem 1 *For every constant $k \geq 2$ there exist a k-database honest-user-SPIR scheme, $\mathcal{B}_k'$, of communication complexity $\mathcal{O}(n^{1/(2k-1)})$.*

4.4 Cube Schemes with Respect to Dishonest Users

The previous section dealt with the SPIR scheme with an honest but curious user. Generally, a dishonest user can cheat in two ways in the previous honest-user-SPIR scheme: in sharing of its index, and by sending invalid queries invalid queries in the original PIR scheme. Here will be described how the scheme from previous section can be made resistant also to dishonest users.

Theorem 2 *There exist a 2-database SPIR scheme, $\mathcal{B}_2''$, of communication complexity $\mathcal{O}(\log n \cdot n^{1/3})$.*

Differences to the $\mathcal{B}_2'$ scheme are that the user independently shares binary representation of the index components i_m, $m = 1, 2, 3$ instead of sharing characteristic vectors. This is done by picking random $(\log_2 l)$-bit strings r_m^0, r_m^1 such that $r_m^0 \oplus r_m^1 = \text{bin}(i_m)$, where $\text{bin}(i_m)$ denote the $(\log_2 l)$-bit binary representation of i_m ($l = n^{1/3}$ is assumed to be a power of 2).

The databases share in addition a random bit s. The bit s is disclosed subject to the condition $\bigwedge_{m=1}^{3}(S_m^0 \oplus S_m^1 = \{r_m^0 \oplus r_m^1\})$ which validates the user's queries. This condition can be verified by a Boolean formula of size $\mathcal{O}(l \log l)$. The bits $w_{000} \oplus s$ and w_{111} are sent in a plain form. The each bit of the rest of the PSM message strings w_σ ($\sigma \neq 000, 111$) is disclosed subject to the codition which is for $\sigma = 100, 011$, $r_1^0 \oplus r_1^1 = \text{bin}(j)$, for $\sigma = 010, 101$, $r_2^0 \oplus r_2^1 = \text{bin}(j)$, and for $\sigma = 001, 110$, $r_3^0 \oplus r_3^1 = \text{bin}(j)$. Each such condition can be verified by a Boolean formula of size $\mathcal{O}(\log l)$.

In reconstruction stage the honest user can reconstruct s and the 8 bits corresponding to index i and compute their exclusive-or to obtain x_i. The scheme's data privacy, relative to any user, follows from the following observations. The user can only obtain from each 6 l-bit strings w_σ a single bit b_σ of w_σ, corresponding to the appropriate shared index component. Thus the user can only learn $(s \oplus b_{000} \oplus b_{111} \oplus b)$, where $b = \bigoplus_{\sigma \neq 000,111} b_\sigma$. If the user's queries are inconsistent, then the user obtains no information on s, and hence (by previous observation) no information at all.

From the sizes of the Boolean formulas used as disclosure condition it follows that the scheme meets the specified complexity bound $\mathcal{O}(\log n \cdot n^{1/3})$. Theorem 2 is generalized by the following theorem which is one of main results of Gertner et al. (see proof from [2], Appendix B).

Theorem 3 *For every constant $k \geq 2$ there exist a k-database SPIR scheme, $\mathcal{B}_k''$, of communication complexity $\mathcal{O}(\log n \cdot n^{1/(2k-1)})$.*

4.5 Necessity of Shared Randomness

Suppose that the databases are allowed to use *private* randomness in answering the user's queries, but they are not allowed to interact without the mediation of the user (and in particular they are not allowed to share a random string unknown to the user). In this setting is now argued that (informatic-theoretic) SPIR cannot be implemented at all, regardless of its complexity, even when the user is honest.

Claim 1 *There exist no (multi-round) k-database SPIR scheme without interaction between the databases, even if the databases are allowed to hold private, independent random inputs, and the user is honest.*

The strong privacy requirement implies that any single database $\mathcal{DB}_j$ cannot respond to the user's queries in a way that depends on the data string x. Formally, at any round the distribution of $\mathcal{DB}_j$'s answer given the previous communication cannot depend on x. For otherwise, this answer distribution must either not follow from a single bit x_i, thus violating the data-privacy requirement, or alternatively reveal to $\mathcal{DB}_j$ the index i on which it depends, thus violating the user's privacy. The independence of private random inputs held by different databases implies that given previous communication the answers of different databases must be independently distributed. Combining the observations made above we have that the joint distribution of all k answers given previous communication is independent of x. Fixing an index i, it follows by induction on the number of rounds that for any $w > 0$ the accumulated communication in the first w rounds is distributed independently of x. This implies that the user's output cannot depend on the value of x_i, contradicting the correctness requirement. [2]

5 Quantum Results

The main result of Kerenidis and Wolf [4] is that honest-user quantum SPIR schemes exist even in the case where the servers do not share any randomness. As proved above, such honest-user SPIRs without shared randomness are impossible in the classical world. This gives another example of a cryptographic task that can be performed with information-theoretic security in the quantum world but that is impossible classically (key distribution [6] is the main example of this). The communication complexity of their k-server QSPIR schemes is of the same order as that of the best known classical k-server PIR schemes.

5.1 The Quantum SPIR scheme

The honest-user QSPIR schemes introduced in [4] work on top of the classical PIR schemes. They have designed them to work particulary on top of the PIR schemes developed by Beimel et al. [7], but those PIR schemes work similarly as all the others known, they just have the best known communication complexity. Here underlying classical PIR scheme is considered as a black box.

The user picks a random string r, and depending on index i and r, picks k queries $q_1, \dots, q_k \in \{0,1\}^t$. In addition, he picks k random strings $r_1, \dots, r_k \in \{0,1\}^a$. The user also holds strings $b_1, \dots, b_k \in \{0,1\}^a$ which are determined by i and r in a way that

$$\sum_{j=1}^{k} a_j \cdot b_j = x_i \qquad (\text{mod } 2),$$

where $a_1, \dots, a_k \in \{0,1\}^a$ are answer strings of servers. The user defines $r'_j = r_j - b_j$ and sets up the following $(1 + k(t+a))$-qubit state

$$\frac{1}{\sqrt{2}}|0\rangle|q_1, r_1\rangle \dots |q_k, r_k\rangle + \frac{1}{\sqrt{2}}|q_1, r'_1\rangle \dots |q_k, r'_k\rangle.$$

The user keeps the first qubit to himself, and sends the other $(t+a)$ qubits to the respective servers. The jth server sees a mixed state of $|q_j, r_j\rangle$ and $|q_j, r'_j\rangle$ which means that the state of that quantum system is not completely known. The density operator language would be right tool to describe such a system but that is out of scope of this work. The jth server performs the following unitary mapping

$$|q_j, r\rangle \to (-1)^{a_j \cdot r}|q_j, r\rangle,$$

which is done for adding phase of the system. Note that server has to learn q_j, r_j and r'_j classically that it can get the answer a_j and perform the mapping. The servers then send all the qubits they have back to the user. The overall communication is then $2k(t+a)$ qubits. The user now has the state

$$\begin{aligned} &\frac{1}{\sqrt{2}}(-1)^{a_1 \cdot r_1}|q_1, r_1\rangle \quad \dots \quad (-1)^{a_k \cdot r_k}|q_k, r_k\rangle \\ +&\frac{1}{\sqrt{2}}(-1)^{a_1 \cdot r'_1}|q_1, r'_1\rangle \quad \dots \quad (-1)^{a_k \cdot r'_k}|q_k, r'_k\rangle. \end{aligned}$$

The common factor $(-1)^{\sum_j a_j \cdot r_j}$ can be ignored because it is global phase and it has no observable effects. Thus previous equals to

$$\begin{aligned} &\frac{1}{\sqrt{2}}|0\rangle|q_1, r_1\rangle \quad \dots \quad |q_k, r_k\rangle \\ +&\frac{1}{\sqrt{2}}|1\rangle(-1)^{\sum_{j=1}^{k} a_j \cdot b_j}|q_1, r'_1\rangle \quad \dots \quad |q_k, r'_k\rangle = \\ &\frac{1}{\sqrt{2}}|0\rangle|q_1, r_1\rangle \quad \dots \quad |q_k, r_k\rangle \\ +&\frac{1}{\sqrt{2}}|1\rangle(-1)^{x_i}|q_1, r'_1\rangle \quad \dots \quad |q_k, r'_k\rangle. \end{aligned}$$

The user can get $|x_i\rangle$ from this by returning everything except the first qubit to 0 by using zero operator which maps any vector to zero vector, and then applying Hadamard transform to the first qubit. Hadamard transform operator is

$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Now the qubit $|x_i\rangle$ needs to be measured in computational basis that the user can learn classical bit x_i. He can't learn anything else, since various states during the protocol never depend on any other x_j. Note also that nowhere in the protocol the servers have shared randomness. Plugging this in the best known classical PIR schemes, due to [7], gives

Theorem 4 *For every $k \geq 2$, there exists a honest-user QSPIR (without shared randomness) with communication complexity* $n^{\mathcal{O}(\log\log(k)/k\log(k))}$.

6 Conclusions

We have gone over some specific classical PIR schemes and how they can be transformed into honest-user SPIR schemes and finally how they can be made resistant to any kind of user behaviour. Classical information theoretic SPIR schemes were proved to be impossible without shared randomness among servers. Allowing user and servers have quantum computers and possibility to communicate with quantum bits made honest-user SPIR schemes possible without shared randomness.

References

[1] B. Chor et al. *Private Information Retrieval.* Journal of the ACM, Vol. 45, No. 6, p. 965–982, 1998.

[2] Y. Gertner et al. *Protecting Data Privacy in Private Information Retrieval Schemes.* STOC 1998.

[3] Y. Gertner et al. *Protecting Data Privacy in Private Information Retrieval Schemes.* Journal of Computer and Systems Sciences, 60(3):592–629, 2000. Earlier version in STOC 98.

[4] I. Kerenidis, R. de Wolf. *Quantum Symmerically-Private Information Retrieval.* arXiv:quant-ph/0307076, 2003.

[5] M. A. Nielsen, I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[6] C. H. Bennett, G. Brassard. *Quantum Cryptography: Public Key Distribution and Coin Tossing.* Proceedings of the IEEE International Conference on Cumputers, Systems and Signal Processing, p. 175–179, 1984.

[7] A. Beimel et al. *Breaking the $\mathcal{O}(n^{1/(2k-1)})$ barrier for information-theoretic Private Information Retrieval.* Proceedings of 43rd IEEE FOCS, p. 261–270, 2002.