On the Private Information Retrieval of the Fragment of the Single-Database

Vladimir B. Balakirsky*

Anahit R. Ghazaryan
School no. 21
Ministry of Defense of Russian Federation
Yerevan, Armenia
e-mail: a_ghazaryan@rambler.ru

ABSTRACT

Private information retrieval schemes are cryptographic protocols designed to safeguard the privacy of database users. In this paper we propose a network-type scheme of private retrieval of the fragment of the single-database. The database user in our scheme is replaced with two users, the user-sender and the user-receiver. As a result of communication, the user-receiver becomes informed about bits of the fragments of the database. If the length of the fragment is not very small, we design a data processing algorithm, where the communication and computational complexities are expressed as functions of the logarithm of the database size. Different extensions and applications of the presented algorithms are discussed.

Keywords

Privacy, Cryptography, Networking, Authentication, Biometrics

1. INTRODUCTION AND THE UNDERLYING PROBLEM

Public databases are an indispensable resource for retrieving up-to-date information. But they also pose a significant risk to the privacy of the database user, since a malicious database owner can follow the queries of the user and infer what the user is after. Private information retrieval schemes allow the database users to retrieve records from public databases without revealing to the database owners which information was retrieved. Private information retrieval problem was intensively studied; in particular, the surveys [2], [3], [4] contain many references.

Formalization of this problem belongs to Chor, Goldreich, Kushilevitz and Sudan [5]. The conventional statement of the private information retrieval problem can be presented as follows. There is a user and either one server or several non-communicating servers with identical copies. The user wants to know the bit x_i of the vector $\boldsymbol{x} = (x_0, \ldots, x_{K-1}) \in \{0,1\}^K$, owned by the server(s), in such a way that each server is ignorant about the index i from the query of the user. A trivial solution, which preserves privacy of the user, is the following: the server(s) transmit the whole vector $\boldsymbol{x}$ to the user. This solution is very expensive because in this case the communication complexity, measured as the number of bits transmitted between the user and the server(s), is K. Private information retrieval protocols allow the user to retrieve data from public databases with smaller communication than just downloading the entire database. Chor at al. [5] showed that in the one-server setup if information-theoretic privacy is required, then there is no better solution than the trivial one when the server transmits the whole database to the user. However, if constraints on the privacy of the retrieval are relaxed, there are many algorithms based on the use of computationally hard problems that have to be solved by the server to discover the data and the use of the so-called one-way hash functions [3], [6]. An implementation of these algorithms is usually time-consuming for the user. On the other hand, encoding of the index i and the content of the database leads to solutions to the multi-server private information retrieval problem [1], [5], [7], although organization of a multi-server scheme where each server must have exactly the same database, can meet technical difficulties.

In [1] we presented a simple algebraic solution to a variant of the multi-server private information retrieval problem, where the user is replaced with two users, the user-sender, who sends the query to the servers, and the user-receiver, who decodes the retrieved bit. In this paper we extend the scheme presented in [1] to the one-server scheme and show that its direct use for this scheme is not possible. In our paper we consider the network with three participants: the user-sender, the user-receiver and the server (see Figure 1). The server has a binary vector

$$\boldsymbol{x} = (x_0, \ldots, x_{K-1}) \in \{0,1\}^K,$$

which is interpreted as content of the database. The user-sender chooses a vector $\boldsymbol{i} = (i_1, \ldots, i_T)$ whose components are T different indices

$$i_1, \ldots, i_T \in \{0, \ldots, K-1\}.$$

The user-receiver wants to know bits $x_{i_1}, \ldots, x_{i_T}$, while the server has to be ignorant about $i_1, \ldots, i_T$. The network-type information retrieval, when the user is split in the user-sender and the user-receiver, brings many additional possibilities. In particular, "the server" can be a target that is en-lighted from some point by the user-sender according to a fixed protocol and broadcasts the replicas. The decision associated with the content of the corresponding fragment is made by the user-receiver, who does not emit energy and his location cannot be discovered. This scheme is close to military-type schemes, where the target is en-lighted from some point according to a fixed protocol, and the decision is made at some other point, which does not

*Deceased

emit energy and cannot be discovered. Furthermore, the query is randomized by the user-sender, but parameters of the randomization should not be delivered to the user-receiver, which decodes the retrieved bits without knowledge of the query. Namely, the randomly chosen matrices $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_T$ are used by the user-sender only for cryptographic purposes and they are not required for decoding the bits $x_{i_1}, \ldots, x_{i_T}$, i.e., the user-receiver decodes these bits without the query. Also, a randomly chosen permutation over components of the query and positions of bits in the fragment introduces essential difficulties for the server, who wants to decode indices $i_1, \ldots, i_T$.

If the user-sender and the user-receiver coincide and $T = 1$, then our setup is the same as the one-server private information retrieval. However, the presented algorithm brings the solution only for the case when T is not very small. This solution is obtained by combining the known approaches developed for multi-server schemes with encoding over positions of the fragment, which can be also understood as interleaving or shuffling of the components of the queries constructed for different positions. Namely, in our scheme the fragment of the database is retrieved using the extension of the multi-server scheme with the randomly chosen permutation over components of the query and positions of bits in the fragment. Our considerations are also relevant to the case, when $T = 1$ and components of the query are sent to L servers. In this case, the setup coincides with the one in [7] and the ideas included in the solution (presented in [1]) are similar, but we believe that our algorithms are essentially simpler for implementation.

Let us also mention an interesting application of information retrieval algorithms for biometric authentication. The particular biometric measurements of a person can be converted to a binary vector containing bits at positions of the so-called significant parameters that are very stable under the observation noise [8], [9]. Therefore, if the person replaces the server in our considerations, then the user-sender can check these bits in some order at the verification stage, and this procedure is equivalent to asking the password of the person. If the order varies in time, then we have a solution when the same biometrics of the person creates many different passwords that can be used for authentication. Furthermore, privacy properties of the retrieval lead to a biometric authentication scheme where the person himself does not know his current password.

2. ALGORITHMIC DESCRIPTION OF THE SOLUTION

2.1 The (n, w)-encoding of indices and polynomial representation of the database

We design a scheme parameterized by the quadruple of integers (m, L, w, n), chosen in such a way that

$$L \le \varphi(m), \quad w < L, \quad \binom{n}{w} \ge K, \tag{1}$$

where $\varphi(m) \approx \frac{2^m-2}{m}$ is the number of cyclotomic classes of the m-ary extension of the Galois field $GF(2)$, denoted by $GF(2^m)$ of length m. We also assume that $T > L$.

Let $\mathcal{J}$ be the set of column-vectors $\boldsymbol{j}$ whose components, denoted by $j_0, \ldots, j_{w-1}$, belong to the set $\{0, \ldots, n-1\}$ and $j_{\tau+1} \ge j_\tau + 1$ for all $\tau = \overline{0, w-2}$. Let us construct a one-to-one mapping

$$k \in \{0, \ldots, K-1\} \leftrightarrow \boldsymbol{j}(k) \in \mathcal{J}, \tag{2}$$

which will be referred to as the (n, w)-encoding. This can be done using the lexicographic algorithm below.

- For $k = 0$, set $j_\tau(0) = \tau$ for all $\tau = 0, \ldots, w-1$. Output the vector $\boldsymbol{j}(0) = (0, \ldots, w-1)$ and the set $\mathcal{J}(0) = \{0, \ldots, w-1\}$.
- For all $k = 1, \ldots, K-1$, set $j_w(k-1) = n$ and
  - find the minimum index $\tau^* \in \{0, \ldots, w-1\}$ such that $j_{\tau^*+1}(k-1) > j_{\tau^*}(k-1)+1$. Set
  $$j_\tau(k) = \begin{cases} \tau, & \text{if } \tau < \tau^* \\ j_\tau(k-1)+1, & \text{if } \tau = \tau^* \\ j_\tau(k-1), & \text{if } \tau > \tau^* \end{cases}$$
  for all $\tau = 0, \ldots, w-1$,
  - output the vector $\boldsymbol{j}(k) = (j_0(k), \ldots, j_{w-1}(k))$ and the set
  $$\mathcal{J}(k) = \{\, j_0(k), \ldots, j_{w-1}(k) \,\}.$$

Let $GF(2^m)$ be the m-ary extension of the Galois field $GF(2)$ constructed using a primitive polynomial $g(\gamma)$ of degree m and let $\alpha \in GF(2^m)$ be the primitive element defined as the root of the polynomial $g(\gamma)$, i.e., $g(\alpha) = 0$. As each element $\beta \in GF(2^m)$ can be uniquely expressed by a linear combination of the elements $\alpha^0, \ldots, \alpha^{m-1}$, components

$$\mathrm{bin}_0(\beta), \ldots, \mathrm{bin}_{m-1}(\beta) \in \{0, 1\}$$

of the column-vector $\mathrm{bin}(\beta)$, which specifies the binary representation of the element β, are determined by the equality

$$\beta = \sum_{d=0}^{m-1} \mathrm{bin}_d(\beta)\alpha^d = \sum_{d:\, \mathrm{bin}_d(\beta)=1} \alpha^d.$$

Moreover, $\boldsymbol{b} = \mathrm{bin}(\beta) \sim \beta = \mathrm{bin}^{-1}(\boldsymbol{b})$ for all $\beta \in GF(2^m)$ and $\boldsymbol{b} \in \{0, 1\}^m$.

Let $\boldsymbol{Z} = (z_0, \ldots, z_{n-1})$ denote the n-tuple of formal variables. We will use the following polynomial representation of the content of the database [7],

$$F_{\boldsymbol{x}}(\boldsymbol{Z}) = \sum_{k:\, x_k=1} f(\boldsymbol{Z}|k), \quad f(\boldsymbol{Z}|k) = \prod_{j \in \mathcal{J}(k)} z_j. \tag{3}$$

For the case $(K, n, w) = (6, 4, 2)$, the (n, w)-encoding given by (2) and the construction of monomials $f(\boldsymbol{Z}|k)$ for the database given by (3) is illustrated in Table 1.

2.2 Encoding algorithm

Let $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_T \in \{0, 1\}^{m \times n}$ be T randomly chosen $m \times n$ binary matrices. The encoding algorithm is organized on the basis of L binary matrices

$$\boldsymbol{A}_1, \ldots, \boldsymbol{A}_L \in \{0, 1\}^{m \times m},$$

whose columns are binary representations of the first m powers of the elements $\alpha^{h_\ell} \in GF(2^m)$, where

$$(h_1, \ldots, h_{\varphi(m)})$$

are leaders of cyclotomic classes of

$$GF(2^m),\ \ l = 1, \ldots, m,$$

of length m. For $i \in (i_1, \ldots, i_T)$ we also introduce the $m \times n$ binary matrix $\boldsymbol{U}(i) = \left[\begin{array}{cccc} \boldsymbol{u}_0(i) & \ldots & \boldsymbol{u}_{n-1}(i) \end{array} \right] \in \{0,1\}^{m \times n}$, where

$$j \in \mathcal{J}(i) \Rightarrow \boldsymbol{u}_j(i) = \left[\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \end{array} \right],\ j \notin \mathcal{J}(i) \Rightarrow \boldsymbol{u}_j(i) = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \end{array} \right] \tag{4}$$

for all $j = 0, \ldots, n-1$. Thus, the 0-th row of the matrix $\boldsymbol{U}(i)$ has weight w and all entries of rows $1, \ldots, m-1$ are zeroes. The encoding is defined as

$$\boldsymbol{S}_{\ell,t} = \boldsymbol{U}(i_t) \oplus \boldsymbol{A}_\ell \boldsymbol{C}_t,\ \ \ell = 1, \ldots, L,\ t = 1, \ldots, T. \tag{5}$$

Equivalently, if $\boldsymbol{C}_t = [\, \boldsymbol{c}_{t,0}, \ldots, \boldsymbol{c}_{t,n-1} \,]$ and

$$\boldsymbol{S}_{\ell,t}(i) = [\, \boldsymbol{s}_{\ell,t;0}, \ldots, \boldsymbol{s}_{\ell,t;n-1} \,],$$

then $\boldsymbol{s}_{\ell,t,j} = \boldsymbol{u}_j(i_t) \oplus \boldsymbol{v}_{\ell,t;j}$ where

$$\boldsymbol{v}_{\ell,t;j} = \boldsymbol{A}_\ell \boldsymbol{c}_{t,j},\ \ j = \overline{0, n-1}.$$

Let $\boldsymbol{\pi} = (\pi(1,1), \ldots, \pi(L,1), \ldots, \pi(1,T), \ldots, \pi(L,T))$ be a randomly chosen permutation over LT pairs

$$((1,1), \ldots, (L,1), \ldots, (1,T), \ldots, (L,T)).$$

The query is formed as the set

$$\boldsymbol{S}' = \left[\begin{array}{ccc} \boldsymbol{S}'_{1,1} & \ldots & \boldsymbol{S}'_{1,T} \\ & \vdots & \\ \boldsymbol{S}'_{L,1} & \ldots & \boldsymbol{S}'_{L,T} \end{array} \right]$$

consisting of LT matrices defined as follows:

$$\boldsymbol{S}'_{\pi(\ell,t)} = \boldsymbol{S}_{\ell,t} \in \{0,1\}^{m \times n},\ \ \ell = 1, \ldots, L,\ t = 1, \ldots, T. \tag{6}$$

The query $\boldsymbol{S}'$ is transmitted to the server row-by-row.

2.3 Constructing the server's replica to the query

The server runs the following algorithm. For any $\ell = 1, \ldots, L$, $t = 1, \ldots, T$,

- Represents the matrix $\boldsymbol{S}'_{\ell,t}$ by the vector $\boldsymbol{Z}'_{\ell,t}$ of length n whose elements are defined as

  $$Z'_{\ell,t;j} = \mathrm{bin}^{-1}(\boldsymbol{s}'_{\ell,t;j}) \in GF(2^m),\ j = \overline{0, n-1}.$$

- Substitutes the vector $\boldsymbol{Z}'_{\ell,t}$ for the argument $\boldsymbol{Z}$ of the polynomial $F_{\boldsymbol{x}}(\boldsymbol{Z})$, defined in (3), and computes

  $$\boldsymbol{R}'_{\ell,t} = F_{\boldsymbol{x}}(\boldsymbol{Z}'_{\ell,t}) = \sum_{k:\, x_k = 1} \prod_{j \in \mathcal{J}(k)} Z'_{\ell,t;j} \in GF(2^m).$$

- Computes the binary column-vector

  $$\boldsymbol{r}'_{\ell,t} = \mathrm{bin}(\boldsymbol{R}'_{\ell,t}) \in \{0,1\}^{m \times 1}.$$

  Then the server sets

  $$\boldsymbol{R}' = \left[\begin{array}{ccc} \boldsymbol{r}'_{1,1} & \ldots & \boldsymbol{r}'_{1,T} \\ & \vdots & \\ \boldsymbol{r}'_{L,1} & \ldots & \boldsymbol{r}'_{L,T} \end{array} \right]$$

  and transmits $\boldsymbol{R}'$ to the user-receiver.

2.4 Decoding algorithm

The user-receiver constructs the bits $x_{i_1}, \ldots, x_{i_T}$ using the received replicas, the permutation $\boldsymbol{\pi}$ and the binary $m \times mL$ matrix $\boldsymbol{D}$ determined by integers n, w and the primitive polynomial $g(\gamma)$.

The user-receiver runs the following decoding algorithm. For any $t = \overline{1, T}$,

- Constructs the binary column-vector $\boldsymbol{r}_t \in \{0,1\}^{mL \times 1}$ by concatenating column-vectors

  $$\boldsymbol{r}'_{\pi(1,t)}, \ldots, \boldsymbol{r}'_{\pi(L,t)}.$$

- Computes the binary column-vector $\boldsymbol{D}\boldsymbol{r}_t$ of length m and output the decision

  $$\left\{ \begin{array}{lcl} \boldsymbol{D}\boldsymbol{r}_t = \boldsymbol{0}^{(m)} & \Rightarrow & x_{i_t} = 0, \\ \boldsymbol{D}\boldsymbol{r}_t \neq \boldsymbol{0}^{(m)} & \Rightarrow & x_{i_t} = 1, \end{array} \right. \tag{7}$$

  where $\boldsymbol{0}^{(m)}$ denotes the all-zero column-vector of length m.

2.5 Constraints on parameters and complexities of the data processing scheme

The communication complexity of the scheme is defined as the total number of bits transmitted over the channels user-sender $\rightarrow$ server $\rightarrow$ user-receiver. As $mLnT$ and mLT are the sizes of the query and the replica, the communication complexity can be expressed as

$$\mathrm{Comp} = mLnT + mLT.$$

We are also interested in the quantity

$$c = \mathrm{Comp}/(T\lceil \log K \rceil + T),$$

since $T\lceil \log K \rceil$ is the number of bits needed to specify the indices $i_1, \ldots, i_T$ when no constraints on privacy of the retrieval are included. The computational complexity is understood as the total number of arithmetic operations performed by the user-sender, the server and the user-receiver. As it will follow from the description of the scheme, all operations are reduced to matrix multiplications of binary matrices and their number is linear in n and T.

If we use the approximation $\binom{n}{n/2} \approx 2^n$ and the approximation to the function $\varphi(m)$, then $n, L, w \approx \log K$ and $m \approx \log\log K$. Therefore, $\mathrm{Comp} \approx \hat{c} T \log K$, where $\hat{c} = (\log\log K)\log K$.

REFERENCES

[1] V. B. Balakirsky, A. R. Ghazaryan, "Algebraic approaches to a network-type private information retrieval", In: *Emerging Trends of Information and Communication Technologies Security*, B. Akhbar, H. R. Arabnia (Eds.), Elsevier, pages 245–251, 2014.

[2] W. Gasarch, "A survey on private information retrieval", *Bulletin of the EATCS*, pp. 72–107, 2004.

[3] R. Ostrovsky, W. E. Skeith III, "A survey of single-database private information retrieval: techniques and applications", *Lecture Notes on Computer Science*, vol. 4450, pp. 393–411, 2007.

[4] S. Yekhanin, "A locally decodable codes and private information retrieval schemes". In: *Foundations and Trends in Teoretical Computer Science*, vol. 7, no. 1, pp. 1–117, 2011.

[5] B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan, "Private information retrieval", Proceedings of the 36th Annual Foundations of Computer Science, pp. 41–50, 1995; Also, in *Journal of the ACM*, vol. 45, pp. 965–981, 1998.

[6] A. Beimel, Y. Isahai, E. Kushilevitz, T. Malkin, "One-way functions are essential for single-server private information retrieval", *Proc. 31-st Anual ACM Symposium on the Theory of Computing*, 1999.

[7] D. Woodruff, S. Yekhanin, "A geometric approach to information-theoretic private retrieval", *Proc. 20-th IEEE Computational Complexity Conference (CCC)*, pp. 275–284, 2005.

[8] V. B. Balakirsky, A. J. Han Vinck, "Biometric authentication based on significant parameters", *Lecture Notes in Computer Science: Biometrics and ID Management*, vol. 6583, pp. 13–22, 2011.

[9] V. B. Balakirsky, A. J. Han Vinck, "Algorithms for processing biometric data oriented to privacy protection and preservation of significant parameters". In: *New Trends and Developments in Biometrics*, InTech, pp. 303–333, 2012. Online: www.intechopen.com (Subjects; Computer and Information Science; Artificial Intelligence).

$k=$	0	1	2	3	4	5
$\mathcal{J}(k)=$	$\{0,1\}$	$\{0,2\}$	$\{1,2\}$	$\{0,3\}$	$\{1,3\}$	$\{2,3\}$
$f(Z\|k)=$	z_0z_1	z_0z_2	z_1z_2	z_0z_3	z_1z_3	z_2z_3

Table 1: The sets $\mathcal{J}(k)$ and the monomials $f(Z|k)$ for the database specified by a binary vector of length $K=6$, when $w=2$.

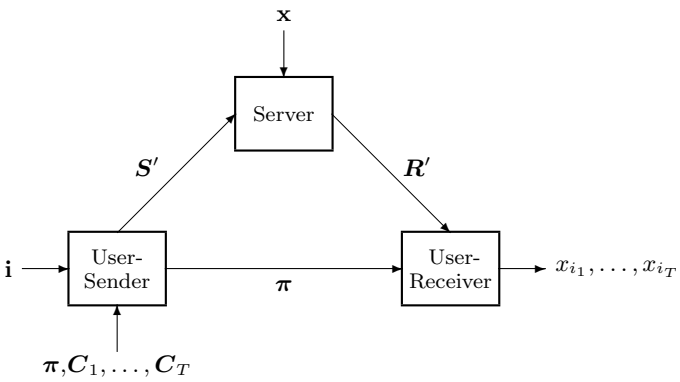

Figure 1. Structure of the data processing scheme.