Algebra and Discrete Mathematics
Volume **19** (2015). Number 2, pp. 302–312

RESEARCH ARTICLE

Towards practical private information retrieval from homomorphic encryption

Dmitry Zhuravlev*

Communicated by V. V. Kirichenko

Abstract. Private information retrieval (PIR) allows a client to retrieve data from a remote database while hiding the client's access pattern. To be applicable for practical usage, PIR protocol should have low communication and computational costs. In this paper a new generic PIR protocol based on somewhat homomorphic encryption (SWHE) is proposed. Compared to existing constructions the proposed scheme has reduced multiplicative depth of the homomorphic evaluation circuit which allows to cut down the total overhead in schemes with ciphertext expansion. The construction results in a system with $O(\log n)$ communication cost and $O(n)$ computational complexity for a database of size n.

Introduction

Confidentiality of queries to publicly accessible on-line data sources is becoming increasingly important for retrieving up-to-date information in many domains, such as patent and media databases, real-time stock quotes, Internet domain names, location-based services, on-line behavioral profiling and advertising, e-commerce and search engines. The *private information retrieval* (PIR) technology allows to protect client's query

*The author would like to thank Ihor Samoilovych for his helpful discussions in the process of this work.

2010 MSC: 11T71.

Key words and phrases: protocols, encryption, servers, complexity theory, private information retrieval, homomorphic encryption.

in such a way that server (or database administrator) cannot infer the purpose of the query while still being able to return the desired data (see Fig. 1).

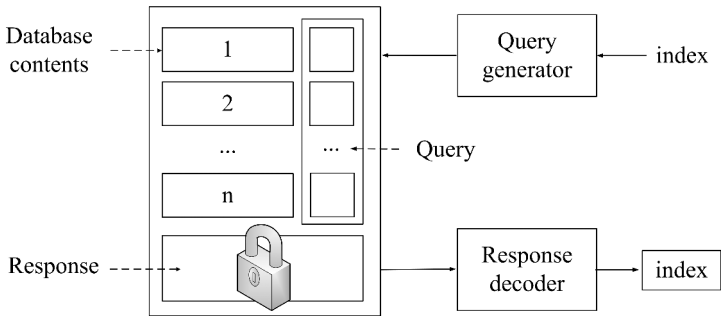

Figure 1. PIR scheme.

PIR was introduced by Chor et al. [1] in non-colluding multi-server settings. In [2] it was proposed the first single database PIR protocol based on hardness computationally assumptions. Namely, the security of the proposed scheme relies on quadratic residuosity problem. The communication complexity of the protocol is $O(2^{\sqrt{\log n \log\log N}})$, where n is the number of bits in the database and N is a composite modulus.

Recent progress in homomorphic cryptography gives rise for new approaches to PIR. In [3] Gentry constructed the first *fully homomorphic encryption* (FHE) scheme — an encryption scheme that supports arbitrary number of additions and multiplications over ciphertexts, and therefore admits to compute arbitrary boolean circuits over encrypted data. In particular, the selection circuit to access the database can be computed over ciphertexts. Hence, one can encrypt the index bitwise and then apply the selection circuit to the encrypted database.

All existing FHE schemes are too expensive to be practical. On the other hand, there exist [4] more practical *somewhat homomorphic encryption* (SWHE) schemes that preserve only limited number of operations. In [5] Brakerski et al. proposed a generic PIR protocol that utilizes a SWHE scheme and a symmetric encryption scheme as building blocks. In their protocol the client uses the symmetric scheme to encrypt the index, and then the server homomorphically decrypts it during query evaluation. Thus, the client's query is short but the server computational cost and response size can be quite large because of the deep response generation circuit.

In [6] Yi et al. constructed a PIR protocol with communication complexity $O(\log n)$ and computational complexity $O(n \log n)$ using the SWHE scheme from [7]. Similar approach is presented in [8] where the SWHE scheme from [9] is exploited to privately retrieve the data. In their protocol a tree-based compression scheme is used to reduce the communication complexity.

Contribution. The presented approach improves both computational and communication costs compared to the previous SWHE based PIR protocols. To adopt PIR for using SWHE, in the proposed protocol the multiplication depth (the number of nested multiplications) of response generation circuit was decreased. For example, a SWHE that can evaluate circuits of depth 5 is sufficient to retrieve the data from a database containing more than 8 billion rows. This multiplication depth is practical for state-of-the-art SWHE [4]. Moreover, the proposed PIR protocol utilizes the recursive retrieval algorithm from [10], which allows to reduce computational complexity from $O(n \log n)$ to $O(n)$. The security of proposed generic protocol is based on the security of the underlying SWHE scheme.

1. Preliminaries

In this section the concepts of PIR protocol and SWHE scheme will be introduced.

1.1. Notations

Throughout the paper we will use the following notation. In the sequel, n denotes the database size in bits and $l = \lceil \log_2 n \rceil$ gives the bit capacity of database indexes. Vector indexes always start from 0, e.g. $a = (a_0, a_1, \ldots, a_{n-1})$. For nonnegative l-bit integer $i = \sum_{k=0}^{l-1} i_{(k)} \cdot 2^k$ we denote $(k+1)$-th bit of i as $i_{(k)}$ for all $k \in \{0, 1, \ldots, l-1\}$.

For all $a \in \{0, 1\}$ we use $a^{\mathcal{R}}$ to denote corresponding additive identity $0_{\mathcal{R}}$ and multiplicative identity $1_{\mathcal{R}}$ of unitary ring $\mathcal{R}$, namely

$$a^{\mathcal{R}} \stackrel{\text{def}}{=} \begin{cases} 0_{\mathcal{R}}, & \text{if } a = 0; \\ 1_{\mathcal{R}}, & \text{othewise.} \end{cases}$$

For some unitary ring $\mathcal{R}$ and all $b = (b_0, b_1, \ldots, b_{l-1}) \in \mathcal{R}^l$ we define associated element to $(k+1)$-th bit of nonnegative l-bit integer t as

$$b_k^t \stackrel{\text{def}}{=} \begin{cases} b_k + 1_{\mathcal{R}}, & \text{if } t_{(k)} = 0; \\ b_k, & \text{othewise.} \end{cases}$$

If $\mathcal{A}$ is a *probabilistic polynomial time* (PPT) Turing machine, by $\Pr\left[\mathcal{A}(x) = y\right]$, we denote the probability that y is equal to answer generated by $\mathcal{A}$ on input x. By $\mathcal{A}^{\mathcal{B}}(\cdot)$, we denote an algorithm that can make oracle queries to $\mathcal{B}$.

1.2. Definitions

We are now ready to define a single database computational private information retrieval. PIR protocols consist of two interactive PPT Turing machines $\mathcal{C}, \mathcal{S}$ which are called the *client* and the *server*, respectively. Each will take as input the security parameter λ and the size of the database n. The server will take as input the database $d = (d_0, d_1, \ldots, d_{n-1}) \in \{0,1\}^n$. The client will take as input an index $i \in \{0, \ldots, n-1\}$, and at the end of the protocol the client will output a bit d_i. This notion can be formally described by the following definition:

Definition 1 (PIR Correctness)**.** A PIR protocol $(\mathcal{C}, \mathcal{S})$ is correct if for any λ, n, i and d as specified above, if

$$(out_{\mathcal{C}}, out_{\mathcal{S}}) \leftarrow \langle \mathcal{C}(1^{\lambda}, n, i), \mathcal{S}(1^{\lambda}, n, d) \rangle$$

then $out_{\mathcal{C}} = d_i$.

One of the most important properties of the private information retrieval protocol is the possibility of non-revealing of the index i to the server. In this paper, we consider a PIR scheme to be secure in the sense that it is computationally infeasible for an adversary to distinguish two queries.

Definition 2 (Negligible function)**.** A positive function μ is *negligible in* λ, or just *negligible*, if for every positive polynomial p and any sufficiently large λ it holds that $\mu(\lambda) \leqslant 1/p(\lambda)$.

Formally the security of PIR protocol is defined as follows:

Definition 3 (PIR Security)**.** A PIR protocol $(\mathcal{C}, \mathcal{S})$ is secure if for all i and j from $\{0, 1, \ldots, n-1\}$ and all non-uniform PPT Turing machines $\mathcal{A}$ there exists a negligible function μ such that

$$\Big| \Pr\left[(out_{\mathcal{C}}, out_{\mathcal{A}}) \leftarrow \langle \mathcal{C}(1^{\lambda}, n, i), \mathcal{A}(state) \rangle : out_{\mathcal{A}} = i\right] - \\ - \Pr\left[(out_{\mathcal{C}}, out_{\mathcal{A}}) \leftarrow \langle \mathcal{C}(1^{\lambda}, n, j), \mathcal{A}(state) \rangle : out_{\mathcal{A}} = i\right] \Big| \leqslant \mu(\lambda) \quad (1)$$

The basis of our PIR protocol is encryption that allows to securely compute polynomial functions of bounded degree.

Definition 4 (Somewhat homomorphic encryption)**.** A symmetric *somewhat homomorphic encryption* (SWHE) scheme is a tuple of three PPT algorithms $\mathcal{E} = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ in which its spaces of plaintexts and ciphertexts are rings and exists a positive number M such that for all polynomial p with $\deg(p) \leqslant M$, all plaintexts $m_0, \ldots, m_k$, and all outputs $sk \leftarrow \mathrm{KeyGen}(1^\lambda)$, we have

$$\mathrm{Dec}\left(sk, p(\mathrm{Enc}(sk, m_0)), \ldots, \mathrm{Enc}(sk, m_k))\right) = p(m_0, \ldots, m_k). \tag{2}$$

We say that the cryptosystem is secure, if the adversary unable to distinguish pairs of ciphertexts based on the message is encrypted by them.

Definition 5 (IND-CPA Security)**.** A symmetric encryption scheme $\mathcal{E} = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ is *indistinguishable chosen plaintext attack secure (IND-CPA)* if for any PPT adversary $\mathcal{A}$, there is a negligible function μ such that

$$\Big|\Pr\left[sk \leftarrow \mathrm{KeyGen}(1^\lambda) : \mathcal{A}^{\mathrm{Enc}(sk, \mathrm{Select}(\cdot,\cdot,0))} = 1\right] - \\ - \Pr\left[sk \leftarrow \mathrm{KeyGen}(1^\lambda) : \mathcal{A}^{\mathrm{Enc}(sk, \mathrm{Select}(\cdot,\cdot,1))} = 1\right]\Big| \leqslant \mu(\lambda), \tag{3}$$

where $\mathrm{Select}(m_0, m_1, b) = m_b$ for $b \in \{0, 1\}$.

2. From SWHE to PIR

In this section we describe the construction of PIR from [5,6,8] based on homomorphic cryptography with computational optimization from [10].

2.1. The basic scheme

Let $d = (d_0, d_1, \ldots, d_{n-1})$ be the client's database, where $d_i \in \{0, 1\}$. We take a SWHE scheme $\mathcal{E} = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ on bits (the message space is residue ring $\mathbb{Z}_2 = \{0, 1\}$), which is used as a "black-box" module. The parameters of the scheme are specified to allow below computations.

The idea of retrieving data is based on the observation that one can algebraically realize the comparison of homomorphically encrypted indexes. Let i be an index of an element in the database d, $0 \leqslant i < n$, where $i_{(k)} \in \{0, 1\}$ are the bits of i. The client encrypts index i bitwise,

$Enc(sk, i) = (c_0, \ldots, c_{l-1})$, $c_i \leftarrow Enc(sk, i_{(k)})$, and sends the result to the server. The server can compare the encrypted address $Enc(sk, i)$ with a given index j, algebraically:

$$e_j = (c_0 + \mathrm{Enc}(j_{(0)}) + \mathrm{Enc}(1)) \cdot \ldots \cdot (c_{l-1} + \mathrm{Enc}(j_{(l-1)}) + \mathrm{Enc}(1)),$$

where $\mathrm{Enc}\,(0)$ and $\mathrm{Enc}\,(1)$ are predetermined encryption of 0 and 1. Since the ciphertext space $\mathcal{R}$ is a ring, $0_{\mathcal{R}}$ and $1_{\mathcal{R}}$ are a valid encryption of 0 and 1.

The homomorphic properties of the scheme $\mathcal{E}$ imply that

$$\mathrm{Dec}\,(sk, e_j) = \begin{cases} 1, & \text{if } j = i; \\ 0, & \text{othewise.} \end{cases}$$

The server computes the auxiliary encrypted choice-bits e_k for every $0 \leqslant k < n$. Then, in order to access the data with encrypted index $End(i)$, the server computes the linear combination over all database

$$r = e_0 \cdot \mathrm{Enc}\,(d_0) + \ldots + e_{n-1} \cdot \mathrm{Enc}\,(d_{n-1}), \tag{4}$$

where $\mathrm{Enc}\,(d_i) = d_i^{\mathcal{R}}$ is the deterministic bit encryption. The client decrypts the result and gets the requested value

$$\mathrm{Dec}\,(r) = \sum_{k=0}^{n-1} \mathrm{Dec}\,(e_k) \cdot \mathrm{Dec}\,(d_k^{\mathcal{R}}) = d_i.$$

2.2. Efficiency

Direct computation in formula (4) requires approximately $l + n$ homomorphic additions and $l \cdot n$ multiplications with depth $\lceil \log_2 l \rceil$ on the server side per request. In [10] efficient method that combines calculation of e_i and the linear combination (4) was proposed. The main idea of this method is to consequently reduce the database so that at the end there is only one element left which is the correct requested element after decryption. Construct elements

$$f_i = c_0 \cdot (d_{2i}^{\mathcal{R}} + d_{2i+1}^{\mathcal{R}}) + d_{2i}^{\mathcal{R}}, \ i = 0, 1, \ldots, l-1,$$

where addition and multiplication are the homomorphic operations from the encryption scheme. Note that

$$\mathrm{Dec}\,(sk, f_i) = \begin{cases} d_{2i}, & \text{if } \mathrm{Dec}\,(sk, c_0) = 0; \\ d_{2i+1}, & \text{if } \mathrm{Dec}\,(sk, c_0) = 1. \end{cases}$$

Therefore the requested element is the element of the database $(f_0, \ldots, f_{2^{l-1}-1})$ with encrypted index $(c_1, \ldots, c_{n-1})$ after decryption. We can repeat the same construction for the new database and index. After l steps we obtain only one element, which is d_i after decryption. The scheme of the algorithm is shown on Fig. 2.

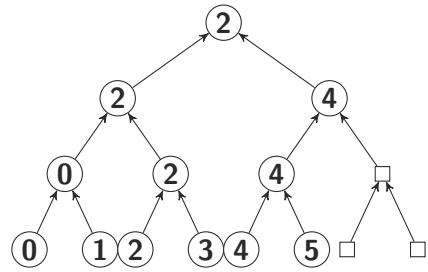

FIGURE 2. Retrieving element with index 010_2 from database of size 6.

This algorithm reduces the number of operations from $O(n \log n)$ to $O(n)$ while calculating server response. Notice that the multiplicative depth of polynomial remains $\lceil \log_2 l \rceil$ as well as in direct computation.

3. Our protocol

In this section, we will show how to decrease the degree of the polynomial in the formula (4) to allow more effective using of server resources.

3.1. PIR with reduced depth

Without loss of generality it is assumed that the database content is represented as an n-bit string $d = (d_0, d_1, \ldots, d_{n-1})$ from which the client wishes to obtain the bit d_i while keeping the index i private. Let $\mathcal{E} = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ be a symmetric SWHE scheme such that its plaintexts form the residue ring $\mathbb{Z}_2$, its ciphertexts form some unitary ring $(\mathcal{R}, +, \cdot)$ and this scheme admits correct evaluation of polynomials of degree $l - 1$, where $l = \lceil \log_2 n \rceil$.

The proposed PIR protocol consists of initialization, query, answering and reconstruction algorithms:

1) $\mathrm{Init}\,(\lambda, n)$: Given a security parameter λ and the database of size n, the client invokes $\mathrm{KeyGen}\,(1^\lambda)$ to generate a secret key sk of the scheme $\mathcal{E}$.

2) QGen $(n, i, 1^\lambda)$: The client encrypts an index i bitwise as $c_k \leftarrow \mathrm{Enc}\,(sk, i_{(k)})$, where $i = i_{(l-1)} \dots i_{(1)} i_{(0)}$ in binary representation. Let $c = (c_0, \dots, c_{l-1})$.

3) RGen (d, c): The server generates and returns a response consisting of two parts, i.e $(r, s) \in \mathcal{R} \times \mathbb{Z}_2$.

- The server computes r in the ring $\mathcal{R}$:

$$r = \sum_{t=0}^{n-1} \left(d_t^{\mathcal{R}} \cdot \prod_{k=0}^{l-1} c_k^t \right) - \sum_{t=0}^{n-1} d_t^{\mathcal{R}} \cdot \prod_{k=0}^{l-1} c_k, \tag{5}$$

where

$$d_i^{\mathcal{R}} \stackrel{\text{def}}{=} \begin{cases} 0_{\mathcal{R}}, & \text{if } d_i = 0 \\ 1_{\mathcal{R}}, & \text{othewise} \end{cases} \quad \text{and} \quad c_k^t \stackrel{\text{def}}{=} \begin{cases} c_k + 1_{\mathcal{R}}, & \text{if } t_{(k)} = 0 \\ c_k, & \text{othewise} \end{cases}$$

- The server computes the sum s of the database elements as elements of the ring $\mathbb{Z}_2$, i.e. $s = \sum_{t=0}^{n-1} d_i$.

4) RExt (r, s, i): Given the response $(r, s) \in \mathcal{R} \times \mathbb{Z}_2$ and index i, the client:

- Decrypts r to obtain $\mathrm{Dec}\,(sk, r) = r' \in \mathbb{Z}_2$
- Computes the bit $d_i = r' + s \cdot \prod_{k=0}^{l-1} i_{(k)} \in \mathbb{Z}_2$.

Theorem 1 (Correctness)**.** *The generic PIR protocol described above is correct for any SWHE sheme on bits $\mathcal{E}$ which can evaluate polynomial of degree $l-1$, any security parameter λ, any database d with any size n and index $0 \leqslant i < n-1$.*

Proof. The degree of polynomial with respect to ciphertext in equation (5) is $l-1$. Thus the homomorphic property of $\mathcal{E}$ implies that

$$\mathrm{Dec}\,(sk, r) + s \cdot \prod_{k=0}^{l-1} i_{(k)} = \sum_{t=0}^{n-1} \left(d_t \cdot \prod_{k=0}^{l-1} {i_{(k)}}_k^t \right) - s \cdot \prod_{k=0}^{l-1} i_{(k)} + s \cdot \prod_{k=0}^{l-1} i_{(k)} = d_i. \ \square$$

3.2. Security proof

Based on the formal definition of security for PIR protocol given in Section 1, we have

Theorem 2 (Security)**.** *If the SWHE scheme $\mathcal{E} = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ is IND-CPA secure, then PIR protocol described above is secure.*

Proof. We show that if a PPT adversary $\mathcal{A}$ against the PIR scheme can distinguish two queries, than there is an adversary $\mathcal{A}'$ that can distinguish two ciphertexts.

Assume $\mathcal{A}$ distinguishes i^0 and i^1 with probability ϵ. That is,

$$\left|\Pr\left[\mathcal{A}(1^\lambda, c^0, state) = i^1\right] - \Pr\left[\mathcal{A}(1^\lambda, c^1, state) = i^1\right]\right| > \epsilon,$$

where $c^i = (c_0^i, \ldots, c_{n-1}^i)$ such that $c_j^k \leftarrow \mathrm{Enc}\left(sk, i_{(j)}^k\right)$. This implies,

$$\begin{aligned}\sum_{k=0}^{l-1} \Big| &\Pr\left[\mathcal{A}(\mathrm{Hyb}\left(c^0, c^1, k\right)) = \mathrm{Hyb}\left(i^0, i^1, k-1\right)\right] - \\ &- \Pr\left[\mathcal{A}(\mathrm{Hyb}\left(c^0, c^1, k-1\right)) = \mathrm{Hyb}\left(i^0, i^1, k-1\right)\right]\Big| \geqslant \\ \geqslant \Big| &\sum_{k=0}^{l-1} \big(\Pr\left[\mathcal{A}(\mathrm{Hyb}\left(c^0, c^1, k\right)) = \mathrm{Hyb}\left(i^0, i^1, k-1\right)\right] - \\ &- \Pr\left[\mathcal{A}(\mathrm{Hyb}\left(c^0, c^1, k-1\right)) = \mathrm{Hyb}\left(i^0, i^1, k-1\right)\right]\big)\Big| > \epsilon,\end{aligned}$$

where $\mathrm{Hyb}\left(a, b, k\right) \stackrel{\mathrm{def}}{=} (a_0, \ldots, a_{k-1}, b_k, \ldots, b_{l-1})$.

Therefore, there must exist $k^\star$ such that

$$\begin{aligned}\Big| &\Pr\left[\mathcal{A}(\mathrm{Hyb}\left(c^0, c^1, k^\star\right)) = \mathrm{Hyb}\left(i^0, i^1, k^\star - 1\right)\right] - \\ &- \Pr\left[\mathcal{A}(\mathrm{Hyb}\left(c^0, c^1, k^\star - 1\right)) = \mathrm{Hyb}\left(i^0, i^1, k^\star - 1\right)\right]\Big| > \frac{\epsilon}{l}.\end{aligned}$$

Consider the algorithm

$$\mathcal{A}'(c^\star) \stackrel{\mathrm{def}}{=} \begin{cases} i_{(k^\star)}^0, & \text{if } \mathcal{A}(1^\lambda, (c_0^0, \ldots, c_{k-1}^0, c^\star, c_{k+1}^1, \ldots, c_{l-1}^1)) = i^0; \\ i_{(k^\star)}^1, & \text{othewise.} \end{cases}$$

$\mathcal{A}'$ runs in polynomial time since $\mathcal{A}$ does. Furthermore, $\mathcal{A}'$ will distinguish $c^\star$ with probability $\frac{\epsilon}{l^2}$. □

4. Efficiency analysis

Since the construction described above can potentially be used with any SWHE scheme, the number of homomorphic operations per request and the size of ciphertexts were used as an efficiency metric.

The most time consuming step of the response generation is computing the linear combination (5). An efficient algorithm of encrypted memory

reading proposed in [10] requires approximately $O(n)$ homomorphic additions and $O(n)$ multiplications on the server side per request.

Current SWHE schemes have a ciphertext expansion property. It means that the size of server response grows with the multiplicative depth of the circuit. Unlike previous construction, in the proposed PIR protocol the multiplicative depth is $\lceil \log_2 (l-1) \rceil$. So overall, the server side computational complexity and communication overhead in our protocol is lower.

The current implementation (based on the SWHE scheme from [9]) of the proposed protocol has the response time 1.88 seconds for a query to a database with 10-bit indexes and 1-Kb records (the benchmarks were measured on an i7 @ 2.20 GHz machine).

Conclusion

This research makes another step towards making *private information retrieval* applicable for the market use-cases. Nowadays the computation and communication overhead are the major issues. We have shown how to improve existing approaches by reducing multiplicative depth of the response generation circuit and utilization recursive retrieval algorithm.

As the future work, we will test our generic protocol with suitable SWHE scheme. Recently, Tian et al. [11] showed that the SWHE scheme from [7] can be effectively realized on GPU. This result gives a way to significantly improve the performance, because a compute-intensive retrieval of multi-bit records admits massively parallel computations. Thus, the speed-up of PIR can be achieved by using parallel computations.

References

[1] B. Chor, E. Kushilevitz, O. Goldreich, M. Sudan, *Private Information Retrieval*, ACM, 45, 1998.

[2] E. Kushilevitz, R. Ostrovsky, *Replication is not needed: single database, computationally-private information retrieval*, In FOCS, 1997, pp 364.

[3] C. Gentry, *A fully homomorphic encryption scheme*, PhD thesis, Stanford University, 2009.

[4] K. Lauter, M. Naehrig, V. Vaikuntanathan, *Can homomorphic encryption be practical?*, Technical Report MSR-TR-2011-61, *Microsoft Research*, 2011.

[5] Z. Brakerski, V. Vaikuntanathan, *Efficient fully homomorphic encryption from (standard) LWE*, FOCS, 2011, pp. 97-106.

[6] X. Yi, M. Kaosar, R. Paulet, E. Bertino, *Single-database private information retrieval from fully homomorphic encryption*, IEEE Trans. Knowl. Data Eng. 25(5), 2013, pp. 1125-1134.

[7] M. Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan, *Fully Homomorphic Encryption over the Integers* Gilbert, H., ed.: EUROCRYPT. Volume **6110** of Lecture Notes in Computer Science, *Springer*, 2010, pp. 24-43.

[8] C. Dong, C. Chen, *A Fast Single Server Private Information Retrieval Protocol with Low Communication Cost*, ESORICS, Lecture Notes in Computer Science, Volume **8712**, 2014, pp 380-399.

[9] Z. Brakerski, C. Gentry, V. Vaikuntanathan, *(Leveled) fully homomorphic encryption without bootstrapping*, ITCS, 2012, pp. 309-325.

[10] D. Zhuravlev, I. Samoilovych, R. Orlovskyi, I. Bondarenko, Y. Lavrenyuk, *Encrypted Program Execution*, TrustCom, 2014.

[11] Y. Tian, M. Al-Rodhaan, B. Song, A. Al-Dhelaan, H. Ma, *Somewhat homomorphic cryptography for matrix multiplication using GPU acceleration*, ISBAST, 2014.

CONTACT INFORMATION

D. Zhuravlev Department of Mechanics and Mathematics
Kyiv National Taras Shevchenko University
Volodymyrska, 64, Kyiv 01033, Ukraine
E-Mail(*s*): dzhuravlev@ukr.net

Received by the editors: 11.03.2015
and in final form 16.07.2015.