Data confidentiality and computations hiding in cloud services for public administration

A. HORUBAŁA[1], D. WASZKIEWICZ[1], M. ANDRZEJCZAK[2], P. SAPIECHA[1]

aleksandra.horubala@gmail.com[1], d.waszkiewicz@tele.pw.edu.pl[1]
michal.andrzejczak@wat.edu.pl[2], sapiecha@tele.pw.edu.pl[1]

[1]Warsaw University of Technology, Faculty of Electronics and Information Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland

[2]Military University of Technology, Faculty of Cybernetics
Urbanowicza 2, 00-908 Warsaw, Poland

Cloud services are gaining interest and are very interesting option for public administration. Although, there is a lot of concern about security and privacy of storing personal data in cloud. In this work mathematical tools for securing data and hiding computations are presented. Data privacy is obtained by using homomorphic encryption schemes. Computation hiding is done by algorithm cryptographic obfuscation. Both primitives are presented and their application for public administration is discussed.

Keywords: public administration, obfuscation, homomorphic encryption.

DOI: 10.5604/01.3001.0012.2001

1. Introduction

Cloud computing is becoming more popular as a way to lower costs of public administration. Several countries has moved selected services to the cloud [1]. Although, there are security concerns linked with cloud services eq. data privacy or physical data localization. Cloud platforms offer security mechanisms, but they are rather organizational than mathematical. Usually cloud servers are physically protected – they are located in secure centers, in isolated networks, often data are encrypted when they are not processed [2], [3], [4]. Unfortunately data owner is not protected from cloud distributors themselves. User has to trust distributors that they would not read, use or sell his data. For sensitive, high value business data these kind of protection cannot be satisfactory. Data privacy independent of the good will of service providers can be obtained by using modern mathematical tools developed in recent years.

In this paper the ideas for homomorphic encryption of public data and cryptographic indistinguishability obfuscation are discussed. This work contains also example of obfuscated cipher with performance results.

2. Homomorphic encryption

First fully homomorphic encryption (FHE) scheme was proposed by Craig Gentry in 2009 in his doctoral dissertation [5]. In recent years a lot new, more efficient, fully homomorphic encryption schemes have been proposed [6], [7]. These schemes usually allow user to execute addition and multiplication on the ciphertexts in a confidential manner. Three of four basic arithmetic operations are able to evaluate: addition, multiplication and subtraction:

$$E_k(m_1) \oplus E_k(m_2) \simeq E_k(m_1 + m_2) \quad (1)$$

$$E_k(m_1) \odot E_k(m_2) \simeq E_k(m_1 \cdot m_2) \quad (2)$$

$$E_k(m_1) \oplus (E_k(-1) \odot E_k(m_2) \simeq E_k(m_1 - m_2) \quad (3)$$

where $E_k(x)$ is encrypted form of x. The main disadvantage of FHE which has to be remembered is lack of operation of division on ciphertexts. No known construction deals with this obstacle.

In FHR adversary cannot learn any partial information about the data, beyond what is explicitly allowed by the leakage function. Homomorphic encryption preserve privacy of data at every stage of the computation.

To perform known algorithms on homomorphically encrypted data, they need to be re-designed to require only the multiplication and addition operations. For example,

Waszkiewicz et al. [8] proposed greedy algorithm for graph vertex cover.

The best known homomorphic encryption schemes are FV, BGV and GSW. Some of these schemes have been implemented in software libraries like SEAL, HElib, FV.

3. Indistinguishability obfuscation

Program obfuscation is the process of making program unintelligible for the user without changing its functionality. Indistinguishability obfuscation is a cryptographic scheme that is computational secure, which can be proven in mathematical way. Program is transformed to convenient form and then encrypted. Implemented modifications cannot be inverted as opposed to mechanical changes made by code obfuscators and precompilers. Two programs having the same input-output behavior are indistinguishable if user having obfuscated version of one of them is not able to decide which of those program was obfuscated.

The main application of program obfuscation is protection of intellectual property. User given access to obfuscated program can only perform operations planned by producer. He cannot examine or modificate program, being not able to understand its structure. No part of the obfuscated program can be decrypted beyond its result.

Known constructions of program obfuscation are based on two cryptographic primitives: fully homomorphic encryption and multilinear maps [9]. Using homomorphic encryption allows one to perform operations on ciphertexts and execute obfuscated programs without their decryption. Application of multilinear maps guarantees that only output of the program can be decrypted (to be exact guessed with overwhelming probability).

To be obfuscated, program needs to be converted to the special form. Every computer program can be represented as a set of boolean formulas, that can be encoded as matrices. Transformed program form consists of the set of pairs of matrices, each pair connected with one binary input variable. Value of the input variable determines which matrix of the pair is going to be used in program execution. Execution of the program consists in multiplying of chosen matrices. All changes listed so far can be easily inverted. To be computationally secure program represented in matrix form needs to be encrypted. Public and private parameters of obfuscation scheme need to be generated. Every element of the matrices is encrypted with private parameters. Encrypted matrices and public parameters form obfuscated program. Obfuscated program can be executed by multiplying encrypted matrices, chosen depend on input values. The result of multiplication of all chosen matrices can be guessed properly with overwhelming probability using public parameters of the scheme. Process of performing program obfuscation and execution of obfuscated program is shown in Figure 1.

Let us notice that during matrix multiplication two operations are performed: addition and multiplication. If data are encrypted with homomorphic scheme these operations can be performed on ciphertexts.

In program obfuscation it is essential that only final output (the result of matrix multiplication) can be guessed. Element of matrices are never encrypted, as well as partial products. These feature is possible due to multilinear maps and special type to cryptographic schemes called graded encoding schemes. Every ciphertext is associated with a level of encryption, which grow with each multiplication. Only after achieving maximal level (after all planed multiplication) ciphertext can be guessed. To introduce levels of ciphertexts, each encrypted element is connected with additional algebraic structure, using technique called encryption over sets [10]. There are two main groups of graded encoding systems: clt13, clt15 by Coron, Tibouchi and Lepoint published in papers [9], [11] (based on homomorphic scheme DGVH, based on integer numbers) and ggh13, ggh15 by Garg, Gentry and Halevi published in papers [12], [13] (based on Gentry's homomorphic scheme [5], based on ideal lattices).

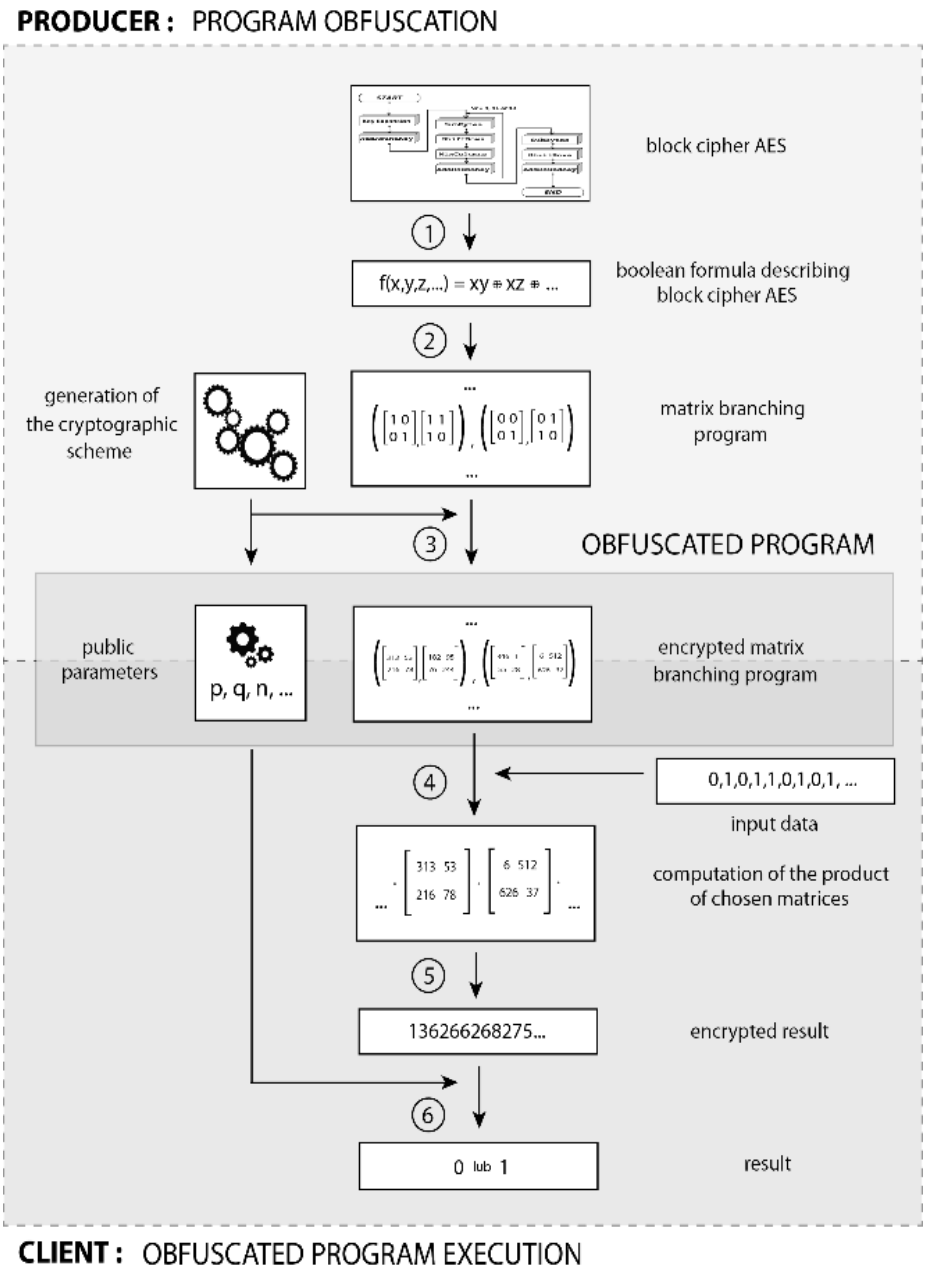

Fig. 1. Obfuscation scheme in general

4. Homomorphic encryption for public administration

In last decade the ongoing digitalization touched nearly every aspect of human life. In the area of public administration plenty of projects aiming at improvising service quality and availability and lowering operational costs have been developed. In Poland there are major flagship projects: ePUAP, eWUŚ, e-Deklaracja, geoportal, CEPiK. Local authorities have also implemented systems allowing to manage citizen's problems via internet without visiting town hall. In Europe there even exist system allowing internet voting in general elections.

Infrastructure costs for maintaining public services are increasing and becoming significant expense for administration. Thus, one of the ways of lowering infrastructure costs is to move to cloud services, what makes sense from a practical and rational economic point of view. That attitude give a lot of concern about data security and privacy, because when data is collected or uploaded from many diverse sources or parties, an online service can host the collection, storage, and computation of and on this data without requiring interaction with the data owner.

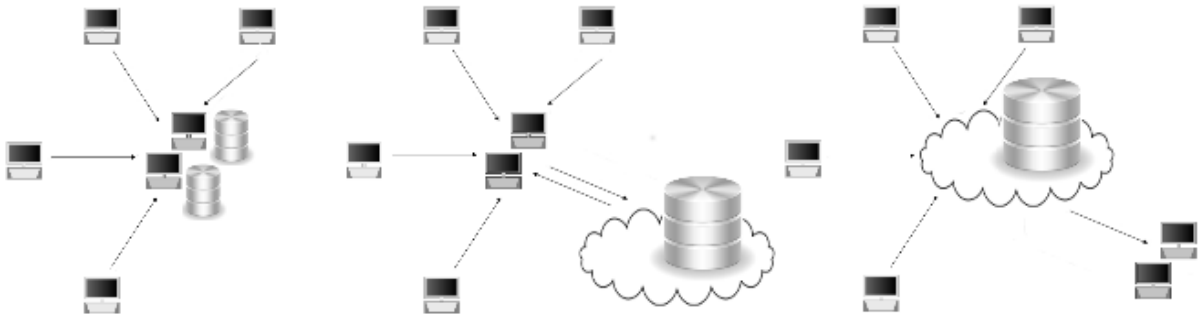

Fig. 2. Paradigms of cloud computations: (left) offline server storage and computation, (middle) cloud servers storage and offline computation, (right) cloud server storage and computation

There are several paradigms of using cloud computation (see fig. 2) as there are several ways for securing cloud solution. One of them is application of property preserving encryption [14], the second one is to use multiparty computation [15]. In most cases cloud security is based on encrypted connection and encrypted stored data [2], [18]. Data is only decrypted when there is need to operate on it. The other way to secure public data is to use homomorphic encryption. Encrypted data could be moved to untrusted cloud environment without any risk of being compromised. Any operation would be done in encrypted form and the encrypted result should be downloaded to local host for decryption (see Figure 3).

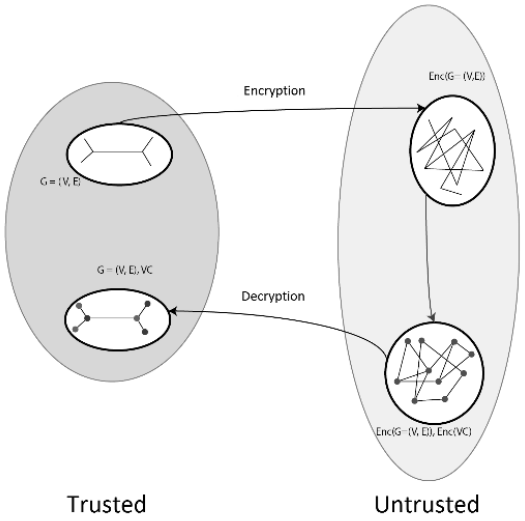

Fig. 3. Execution model for confidential algorithm

During this process no feasible data will be revealed. This homomorphic computation scheme could be used in the following areas:

Medical data: personal medical record could be stored in cloud and accessed by medical units delivering services for chosen person. Whole medical record will be available and exchanged by different authorized service providers. Storing whole medical record in one place can accelerate diagnosis with artificial intelligence algorithms. In this case homomorphic encryption could secure sensitive data and allow to operate on data without any concern about data privacy.

Financial system: financial system has crucial role in everyday life, thus it's security is major issue. Financial information are processed by several different institutions responsible for different areas like health insurance, tax income, retirement funds. Data exchange between these institutions is one of the potential failure point and exposes whole system for additional risk. Moving financial data to the cloud could save taxpayers money by lowering infrastructure maintenance costs, but this must be done with maximum security.

Citizen services: running websites on cloud which allow maintaining citizen information and administrative issues can save citizen time and taxpayers money. There are many issues that can be done without visiting office. Any data leakage can reveal sensitive personal information, so homomorphic encryption could secure data privacy.

There are only several scenarios where public administration can move to cloud services. There are many more other ways to use cloud in public administration.

5. Indistinguishability obfuscation for public administration

There are many applications of program obfuscation. Let us introduce the software producer who wants to protect his intellectual property but also needs to sell his programs.

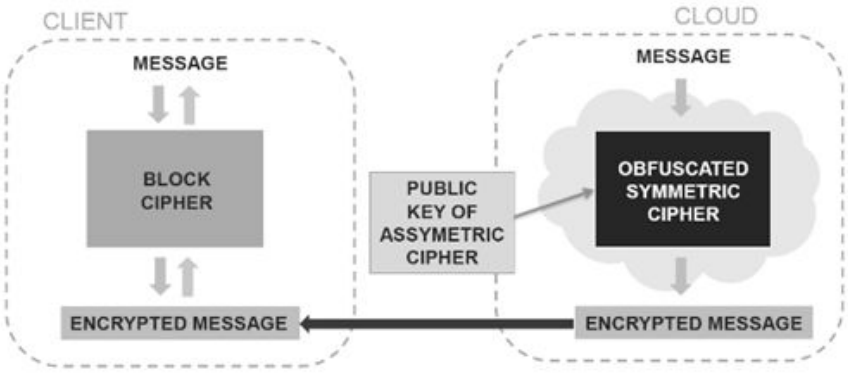

Fig. 4. Creating asymmetric cryptographic scheme from symmetric one

If he sells obfuscated program, everyone can use it but cannot modify it and does not know what they are actually computing (which operations are performed, in which order). The situation is illustrated in the Figure 1. Growth of interest in cloud computing results in growing need of new secure solutions for clients that want to execute their programs on the external servers. If clients put their programs in the cloud in obfuscated form no one will know what they are computing.

This approach can be easily adopted for public administration needs in several scenarios. First scenario describes securing communication channel by creating new asymmetric cryptosystems from symmetric one, which is shown on Figure 4. Obfuscated symmetric cipher can be used as a public key in asymmetric cryptosystem. These will allows to send data or exchange secrets in secure way.

In the other scenario public administration representatives can make computations on public data without leaking information about type of computation. For example obtaining information from simulations of crisis situation or terrorist attack can reveal classified data about procedures and potential weakness. Even the information about this type of simulation can influence on public opinion. In the other case, information about some kinds of simulations can give advantage on real estate market.

6. Obfuscation of Mini-AES

Theory sounds promising, but main question is about performance of proposed solutions. As an example of obfuscation which may be useful in public administration we have chosen and obfuscated Mini-AES cipher. Firstly, Boolean formulas of round cipher are needed. Formulas need to be minimized and presented in optimal form. On the base of these formulas matrix branching program was created. Obfuscation process is presented on Figure 5.

We used implementation of obfuscation created by Malozemoff and published on github [16]. We used variant based on CLT13 graded encoding scheme proposed by Coron, Lepoint and Tibouchi [9]. For creating matrix branching program from Boolean formula we used method of Sahai and Zhandry based on bilinear forms [17] (which we extended in our implementation to multilinear case).

We have performed several experiments on Microsoft Azure cloud server with 32 cores and 448 GB RAM to measure performance of obfuscated two rounds of Mini-AES cipher with pre-compiled rounds. Pre-compiled round means that there are pre-computed matrices for every possible input, so there is no need for matrix multiplication depending on value of input bits in round function. We compared three obfuscation variants: whitebox version with internal key, whitebox version with external key and blackbox version. Internal key means that key is connected and obfuscated with round matrices. External key means that key part is treated as separated obfuscated block. Results can be seen on Table 1. We also examined how security parameter influences sized of ciphertext (and obfuscated program). We chose to perform tests for multiplicative complexity 8. Results are shown in Table 2. Last experiment tested influence of multiplicative complexity on the program size. We set security parameter to 2^{16}. Results are presented in Table 3.

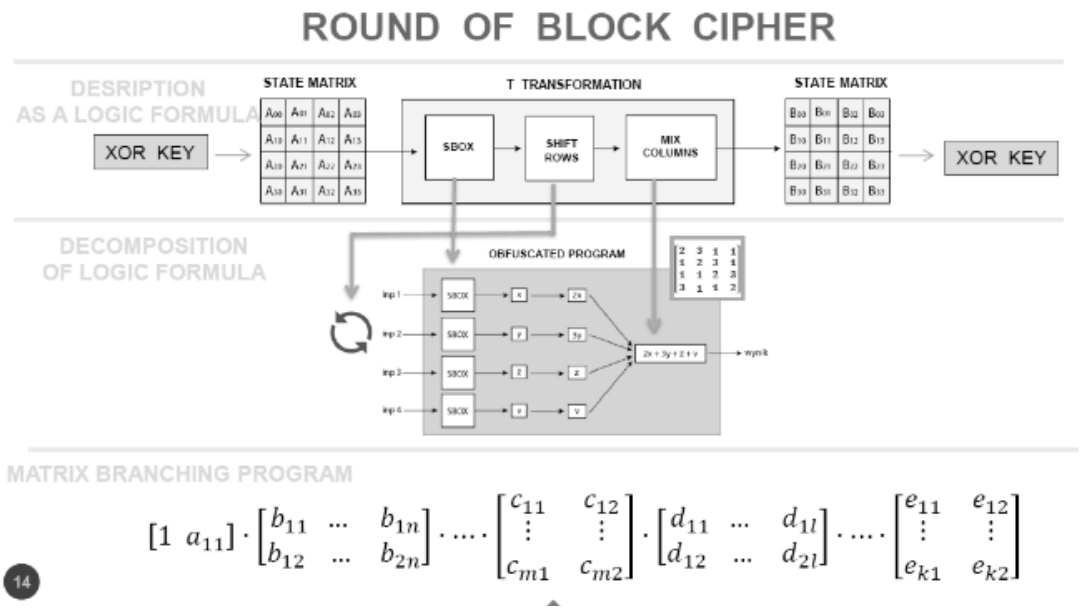

Fig. 5. Obfuscating a round of block cipher

Tab. 1. Results of obfuscation of Mini-AES cipher

2 rounds of Mini AES	whitebox internal key	whitebox external key	Blackbox extrenal key
size of obfuscated program [kB]	11 000	557 000	$\approx 35.7 \cdot 2^{16}$
evaluation time [s]	11	3557	$\approx 13466 \cdot 2^{16}$
multiplicative complexity	9	81	289

Tab. 2. Size of ciphertext on the security parameter for 8 multiplication

security level	2^8	2^{16}	2^{24}	2^{32}
size of single ciphertext [kB]	12.6	58.9	145.1	257.0
speed of growth	–	4.67	2.44	1.88

Tab. 3. Size of ciphertexts depend on multiplicative complexity for security parameter 2^{16}

number of multiplications	4	8	16	32
size of single ciphertext [kB]	58.9	189.3	393.6	672.0
speed of growth	–	3.21	2.07	1.7

7. Conclusions

Program obfuscation and homomorphic encryption are promising techniques that can be used for public administration. They allow one to encrypt public data and perform operations on them in encrypted form, which is secure even in untrusted environment.. Proposed schemes give novel functionalities, but performance of solutions based on these schemes is still too low, even in cloud environment. The aim of ongoing research is to improve performance of presented schemes and decrease their computational complexity.

8. Bibliography

[1] Zaharia-Radulescu A., Radu I., "Cloud computing and public administration: approaches in several European countries", *Proceedings of the International Conference on Business Excellence*, Vol. 11, Issue 1, 739–749 (2017).

[2] http://www.cloud-council.org/deliverables/CSCC-Security-for-Cloud--Computing-10-Steps-to-Ensure--Success.pdf.

[3] https://cloud.google.com/security/.

[4] https://www.spidersweb.pl/2014/07/zabezpieczenia-danych-w-chmurze.html.

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

[6] Brakerski Z., Gentry C., Vaikuntanathan V., "(Leveled) fully homomorphic encryption without bootstrapping", in: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS'12, pp. 309–325, ACM, NY, USA 2012, DOI: 10.1145/2090236.2090262.

[7] Fan J., Vercauteren F., "Somewhat practical fully homomorphic encryption", Cryptology ePrint Archive: Report 2012/144, http://eprint.iacr.org/2012/144, 2012.

[8] Waszkiewicz D., Andrzejczak M., Horubała A., Sapiecha P., "Confidential greedy fraoh algorithms", 17th Central European Conference on Cryptography, June 28–30, 2017, Warsaw, Poland.

[9] Coron J-S., Lepoint T., Tibouchu M., "Practical multilinear maps over the integers", in: *Advances in Cryptology – CRYPTO 2013*, LNCS, Vol. 8042, pp. 476–493, Springer, 2013.

[10] Naveed M., Kamara S., Wright C.V., "Inference attacks on property-preserving encrypted databases", in: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 644–655, ACM, NY, USA, 2015, DOI:10.1145/2810103.2813651.

[11] Coron J-S., Lepoint T., Tibouchu M., "New multilinear maps over the integers", Cryptology ePrint Archive, Report 2015/162, 2015.

[12] Garg S., Gentry C., Halevi S., "Candidate multilinear maps from ideal lattices", in: *Advences in Cryptology – EUROCRYPT 2013*, LNCS, Vol. 7881, pp. 1–17, Springer, 2013.

[13] Gentry C., Gorbunov S., Halevi S., "Graph--induced multilinear maps from lattices", in: *Theory of Cryptography, 12th Theory of Cryptography, TCC 2015, Warsaw, Poland, March 23–25, 2015, Proceedings*, Part II, LNCS, Vol. 9015, pp. 498–527, Springer, 2015.

[14] Liao X., Uluagac S., Beyah R.A., "S-match: Verifiable privacy-preserving profile matching for mobile social services", in: *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, June 23–26, 2014, pp. 287–298. IEEE, 2014, DOI: 10.1109/DSN. 2014.37.

[15] Yao A.C., "Protocols for secure computations", in: *Proc. FOCS 1982*, Washington, DC, USA, pp. 160–164, IEEE Computer Society, 1982.

[16] Malozemoff A., "Implementation of program obfuscation published on github", https://github.com/amaloz/obfuscation.

[17] Sahai A., Zhandry M., "Obfuscating low-rank matrix branching programs", Cryptology ePrint Archive, Report 2014/772, 2014. https://eprint.iacr.org/2014/772.

[18] VMware, SAVVIS, "Securing the cloud. A review of Cloud Computing, Security Implications and Best Practices" (online), VMware White Paper, 2009.

Poufność danych i ukrywanie obliczeń w chmurze obliczeniowej dla potrzeb administracji publicznej

A. HORUBAŁA, D. WASZKIEWICZ, M. ANDRZEJCZAK, P. SAPIECHA

Chmura obliczeniowa zyskuje coraz większą popularność i staje się ciekawą alternatywą do wykorzystania w administracji publicznej. Istnieje jednak wiele obaw co do bezpieczeństwa i prywatności przechowywanych w chmurze danych osobowych. W tej pracy zaprezentowano matematyczne narzędzia zabezpieczania danych oraz ukrywania obliczeń. Prywatność danych uzyskuje się poprzez wykorzystanie szyfrowania homomorficznego, natomiast ukrywanie danych poprzez obfuskację kryptograficzną. Oba prymitywy zostały zaprezentowane oraz omówiono ich zastosowanie dla administracji publicznej.

Słowa kluczowe: administracja publiczna, obfuskacja kryptograficzna, szyfrowanie homomorficzne.