Secure Algorithms for Biomedical Research in Public Clouds

Martin Beck*, V. Joachim Haupt†, Jan Moennich†, Janine Roy†, René Jäkel‡, Michael Schroeder†, Zerrin Isik†
*TU Dresden, Institute of Systems Architecture, Germany
martin.beck1@tu-dresden.de
†TU Dresden, BIOTEC, Germany
michael.schroeder@biotec.tu-dresden.de
‡TU Dresden, Center for Information Services and High Performance Computing (ZIH), Germany
rene.jaekel@tu-dresden.de

Abstract

Algorithms from the biomedical domain have to face a rapid growth of biological data and therefore a rising demand for computing time. The predictive power of such algorithms is also further improving and becomes increasingly interesting for commercial applications. Cloud Computing – as an already established paradigm to elastically allocate computing resources on demand – offers flexible solutions to deal with the increasing request for compute power. However, security concerns remain when valuable research or business data are being processed in a Public Cloud. Herein, we describe – from the application and security perspective – three biomedical case studies from different domains: Patent annotation, cancer outcome prediction, and drug target prediction developed within the GeneCloud project. Our approach is to realize a data-centric security method to be able to compute on encrypted or blinded data in any non-trustworthy environment accessible by the user.

***Index Terms*—Cloud Computing, data security, privacy, text-mining, outcome prediction, drug repositioning**

1 Introduction

Data in life sciences – such as sequence, structural, pharmacological data and biomedical literature – are ever growing and its usage demands appropriate computational resources. Cloud services offer such resources, but require adjusted algorithms and data management. Security concerns rise when valuable research, business data or personally identifiable information is transferred to a Public Cloud. Moreover, resulting data should be protected in such a way, that nobody except from the submitting party is able to evaluate the results.

Several proposals were made over the last years, like the use of trusted platform modules (TPM) for trusted computing [17], effective separation of data depending on the secrecy level [6], the complete use of fully homomorphic encryption [9] or multi-party computation between several non-colluding parties [20].

Nevertheless, these techniques possess advantages and disadvantages, that need to be considered before application. Some methods like homomorphic encryption provide a high security level, but data transformation is extremely time consuming, thus increasing execution time dramatically. In contrast, techniques – like anonymization – have only a small impact on the computational complexity, but do not provide a high level of security. The balance between different security mechanisms and the efficiency of data transformation is shown in Figure 1.

All authors contributed equally to this work.

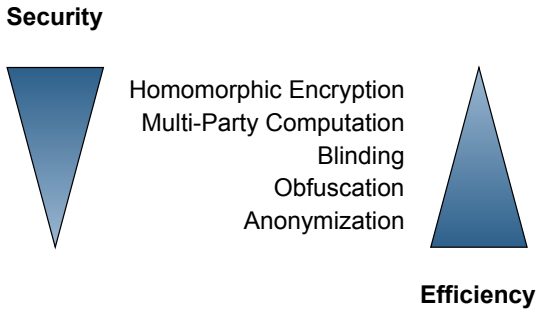

Figure 1. Balance between security and efficiency for different privacy preservation mechanisms

2 Infrastructure

In this section we briefly describe the architectural model as well as the infrastructure used to run Cloud services based on algorithms from biomedical research on a Public Cloud environment. In principal this means to operate on highly vulnerable data from the pharmaceutical domain. Therefore, any Cloud environment used for calculations not under our direct control, e.g. concerning its access rights, is regarded as a potentially non-trustworthy environment.

In our project we have started to construct a Private Cloud solution in order to realize secure versions of the use cases, which allows us to provide services able to compute on encrypted data. Firstly, we have evaluated widely used Open-Source based Cloud middlewares, which operate on top of a virtualization layer. This layer

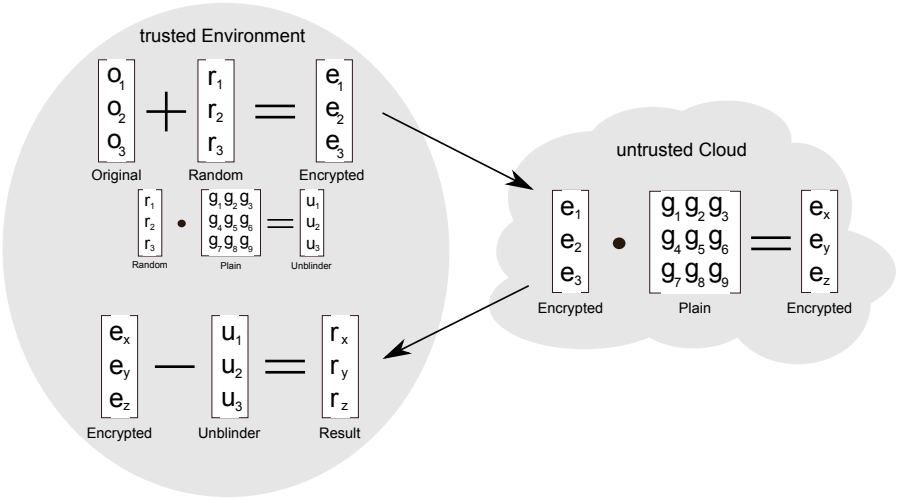

Figure 2. Privacy-preserving matrix-vector multiplication using a blinding technique to hide sensitive information

is accessible via hypervisor instances. In our reference installation we rely on KVM [11], which is a stable project and widely distributed, but we have also examined Xen [19] as a possible alternative. As middleware we have chosen OpenNebula [14] as default, but could also switch alternatively to OpenStack [15] instead. The basic management functionalities of the state-of-the-art Cloud middlewares have evolved towards a rather mature state and provide a broad set of management tools, such as user handling and rights management, full virtual machine (VM) life cycle, monitoring, as well as user and developer interfaces.

Beside the basic functions to interact with the underlying virtualized hardware layer, those middlewares (OpenNebula, OpenStack, Eucalyptus, Nimbus) support elastic scaling of needed computing demands via established quasi-standards, either by using the proprietary EC2 interface (e.g. to submit VMs to the Amazon Cloud) or the open standard OCCI [13].

We are currently evolving the use cases (see sections 4) towards secure Cloud services. The data encryption has to be performed in a fully trustworthy environment, such as our Private Cloud, where we also provide storage for those encrypted data sets. Based on the actual Cloud infrastructure, more complex Cloud models can be realized later on. In the first place, we use the system as a Private Cloud for testing purposes, but since the computing needs are potentially very large, other models, including a fully-grown Hybrid Cloud, acting than as a hybrid Cloud solution, can be realized as well among our partners.

3 Security

The bioinformatics applications, which will be described in the following chapters, utilize very different essential building blocks. Some utilized algorithms are not accessible and must be treated in a black-box setting, while others are using simpler primitives like matrix multiplications. Depending on the actual algorithm, its accessibility and of course privacy requirements regarding the input data, only a subset of available privacy preservation techniques can be applied. We will shortly discuss the three main scenarios of this project regarding possible security solutions over the next sections.

3.1 Secure Text Mining

A very basic requirement to be able to perform text mining is the possibility to compare text or character strings with each other. The input string of one member should however not be known to other participating parties. In case a search over confidential documents is performed, the content of these documents must be protected against access from unauthorized parties.

Beck *et al.* implemented a privacy-preserving string comparison algorithm, which has exactly these requirements fulfilled and is efficient by means of having a linear complexity in the length of the longest compared string [3]. The main parts of this solution are a conversion of the input strings into sets using variable length grams [12], a representation of these sets using Bloom Filters [5] and a privacy-preserving comparison of these representations using additive homomorphic encryption [4].

3.2 Secure Cancer Outcome Prediction

There are two main components of the cancer outcome prediction algorithm presented in 4.2, first a matrix-vector multiplication and second a machine learning algorithm. Several methods were proposed within the security community to securely outsource matrix multiplications to untrusted parties [1, 2]. We use an efficient blinding solution as is shown in Figure 2.

The blinding and all subsequent operations can be done over real numbers, which results in a very efficient protocol with very low computational overhead, but which leaks information about the hidden data, due to the differences in the expected and observed distributions over the input values. Instead of using floating point numbers

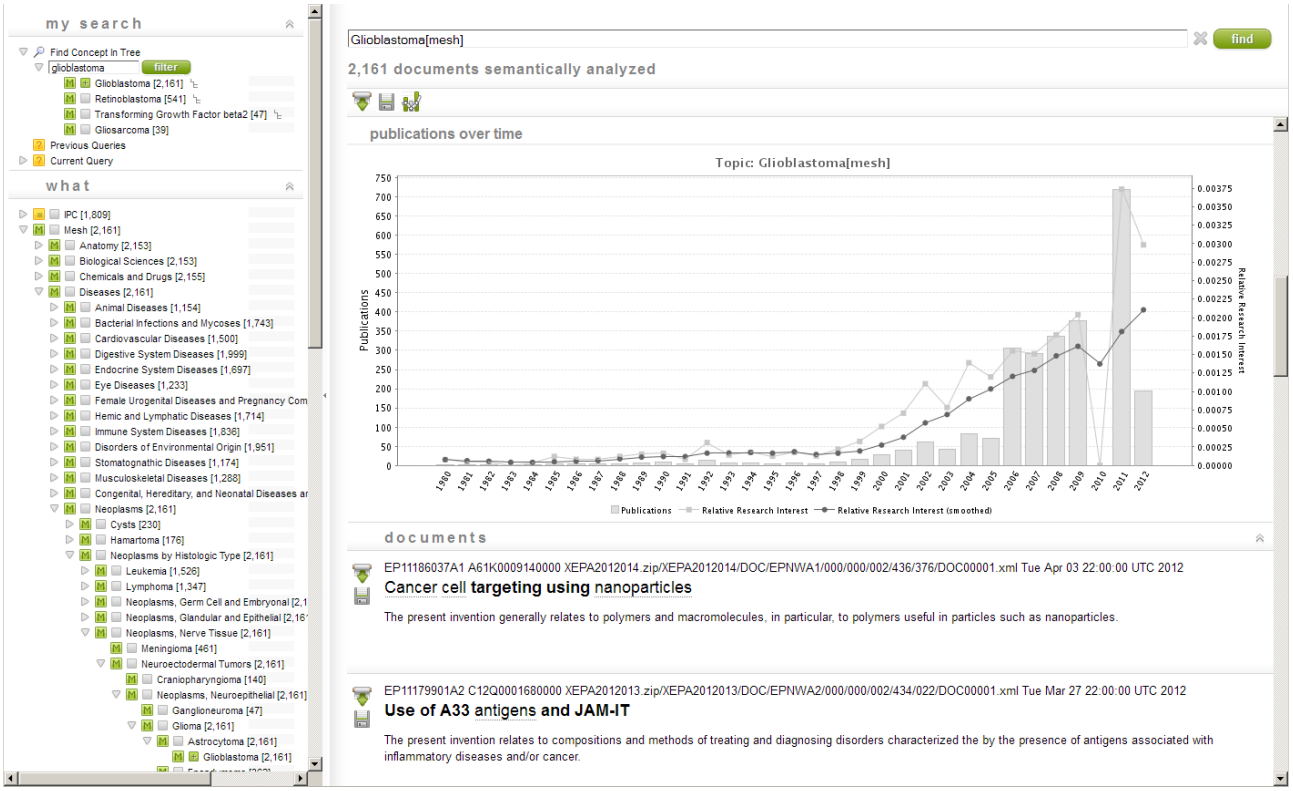

Figure 3. GoPatents: automatic extraction, sorting and statistical evaluation of biomedical entities - Left: Tree representing all assigned document to entities - Upper right: statistical evaluation of the document set - Down right: document set

we use modular arithmetic over a finite prime field $\mathbb{F}_p$. The input values are first converted to integers within this field, using an arbitrary but fixed precision. Blinding is then done by adding a uniformly chosen random element from $\mathbb{F}_p$. Another implemented solution uses an additive homomorphic cryptographic system to encrypt either the matrix or the vector of the multiplication. The actual operation is then performed over the ciphertext, returning an encrypted result, which only the client is able to decrypt.

3.3 Secure Drug Target Prediction

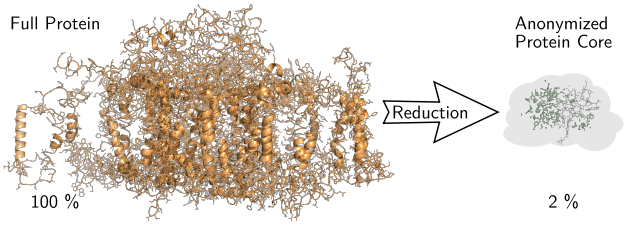

Figure 4. Minimizing and at the same time anonymization of a protein structure to a predefined ligand

Some of the algorithms used for predicting drug targets are not available as source code and thus can only be treated as a black box, which can not be converted directly into a privacy-preserving version. One part of the prediction pipeline, where we find such a black box, is the task to perform a binding site alignment of two protein structures in three dimensional space. The inputs of such an algorithm are descriptions of two protein structures, which contain mainly coordinates with three dimensions, some properties for the referenced atoms and further general descriptions. We strip all the unnecessary information out of the structure to make identification of compared proteins as hard as possible.

Figure 4 shows the result of such a minimization. Only atoms along and around the specified ligand are kept. The atom positions them self are further moved and rotated by some random value to make identification more complex.

4 Use Cases

In this section we describe three domains of biomedical research using big data. This data is at least partially mission-critical, underlies copyright restrictions or contains personally identifiable information and must therefore be protected against illegal access. The large amount of required processing resources, as well as the necessity for privacy protection thus builds a common ground for all of the following use cases.

4.1 Text Mining

The text mining algorithms will be implemented as two use cases.

The first use case is the creation of GoPatents (Figure 3), which automatically analyzes European patents for biomedical entities and sorts them into the IPC ontology. In contrast to the search engine by Doms *et al.* GoPubMed [7] it does not only analyze the abstracts, but

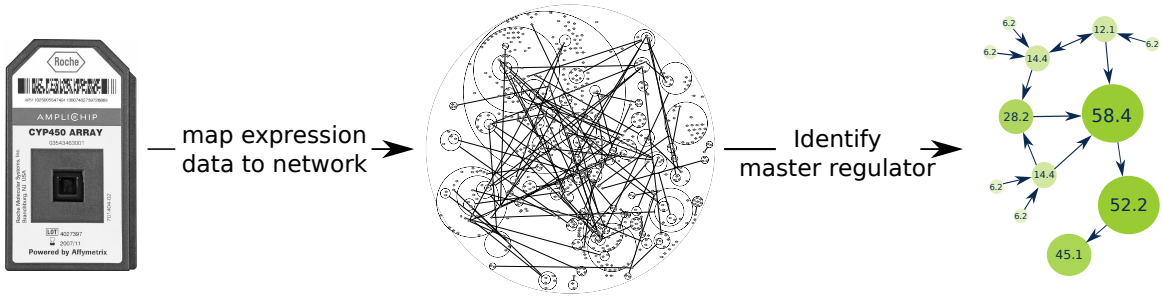

Figure 5. The work flow of network-based biomarker discovery

whole documents. Due to the longer texts, much more processing power is required (approx. 6,000 CPU h for 2 million patents instead of 300 CPU h for 20 million abstracts), hence it requires massive parallel processing to analyze these patents within a rather short time frame. Further the patents have to be protected due to copyright restrictions.

The second use case processes the relations between different entities, such as proteins, drugs and diseases. However, this information is usually not easily accessible from a database, it has to be collected from many different journals. We have developed a system that retrieves a list of statements about the relation between two different entities of the given drugs, proteins, and diseases. After a search for two entities, all documents are automatically fetched and all sentences with both entities are extracted. These sentences are ranked according to their importance using a maximum entropy classifier. While such analysis for combinations of drugs, proteins or diseases is already available, we are developing an online application that identifies all counterparts for one known entity. As these calculations are computationally intensive (approx. 3 CPU months for 22 million Medline abstracts), access to a Cloud environment can shorten the overall execution time considerably by parallel processing of search tasks. Currently, a demonstrator for a few thousand abstracts is available. However, we need to adjust this application for the Cloud environment to run the analysis for the whole Medline database.

4.2 Cancer Outcome Prediction

Cancer outcome prediction aims to address the problem of learning how to forecast disease progression from gene expression and thus allowing refined treatment options. Thus, gene expression measured via microarrays helps to reveal underlying biological mechanisms of tumor progression, metastasis, and drug-resistance in cancer studies. The gene signatures obtained in such analysis could be addressed as biomarkers for cancer progression.

The experimental or computational noise in data and limited tissue samples collected from patients might reduce the predictive power and biological interpretation of such signature genes. Network information (e.g. about protein-protein interactions) efficiently helps to improve outcome prediction and reduce noise in microarray experiments [8]. For this purpose, network information has been integrated with microarray data in various studies in the last decade.

Winter *et al.* developed an algorithm – called NetRank – that employs protein-protein interaction networks and ranks genes by using the random surfer model of Google's PageRank algorithm [18]. Figure 5 shows the general approach of NetRank. Firstly, gene expression values measured by microarrays are mapped to network data, which might be either physical protein-protein interaction or regulatory information. Afterward, NetRank is applied on the network to identify master regulators based on gene expression data and network information. Hence, the algorithm provides an integration of the topological information (i.e. connectivity and random walk) and microarray data (i.e. node score) to explore crucial signature genes for outcome prediction.

The performance of the algorithm was evaluated in two studies. In the first study, NetRank was applied on gene expression data obtained from 30 pancreas cancer patients and seven candidate biomarker genes could be identified that are able to predict the survival time of patients after tumor removal with 70% accuracy [18]. The second study was a systematic assessment of network-based outcome prediction on 25 cancer data sets, where the authors could show the general applicability of the algorithm on different cancer types [16].

Due to the effectiveness of the approach, implementation in the Cloud as a publicly available software could accelerate as well as simplify cancer outcome prediction.

4.3 Drug Target Prediction from Binding Site Similarity

Drug repositioning applies established drugs to new disease indications with increasing success. A prerequisite for drug repositioning is drug promiscuity (polypharmacology) – a drug's ability to bind to several targets. There is a long standing debate on the reasons for drug promiscuity. Based on large compound screens, hydrophobicity and molecular weight have been suggested as key reasons. However, the results are sometimes contradictory and leave space for further analysis. Protein structures offer a structural dimension to explain promiscuity: Can a drug bind multiple targets because the drug is flexible or because the targets are structurally similar or even share similar binding sites?

Haupt *et al.* contributed to the discussion of causes of

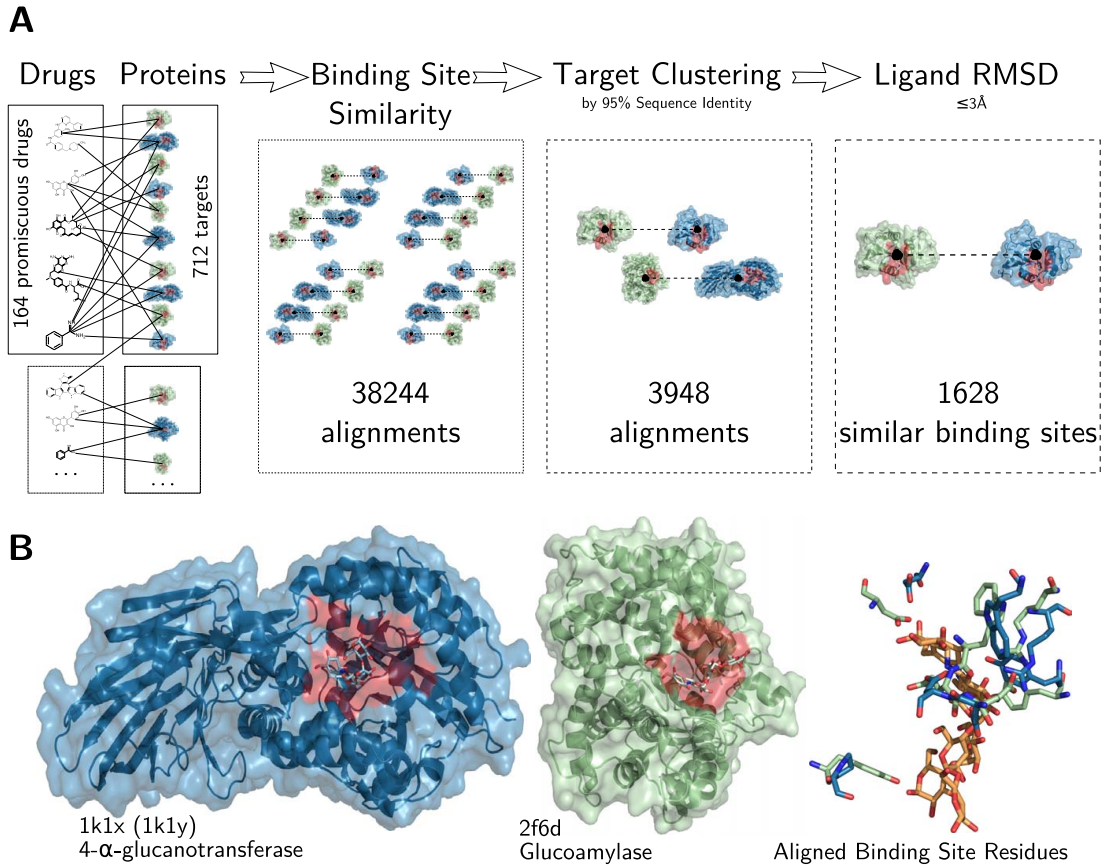

Figure 6. (A) The work flow resulting in 1628 similar binding site pairs with well superposed ligands. (B) Similar binding sites in a pair of proteins targeted by acarbose. The two bound structures of acarbose are shown as orange sticks on the right with superposed binding site residues. Figure adapted from [10]

drug promiscuity by pursuing a comprehensive structural approach and by investigating two more possible sources for drug promiscuity: ligand flexibility and binding site similarity [10]. However, the drug data set is limited in size due to the restriction to PDB. They analyzed a structural data set of 164 promiscuous drugs bound to 712 unique protein targets from the PDB (Figure 6.A). Regarding drug physicochemical properties, they found no correlation between hydrophobicity or molecular weight and the degree of promiscuity of a drug in contrast to other studies on large screening libraries. Subsequently, they clustered the drug conformers and compared all binding sites of a drug. As a result, they found a weak correlation of the degree of drug promiscuity to ligand flexibility (r = 0.2), a correlation to structural similarity (r = 0.76) and even higher to the number of similar binding sites (r = 0.81, example in Figure 6.B). Furthermore, they found that for 71 % of the drugs at least one pair of their targets' binding sites is similar and for 22 % all are similar. Thus, they conclude that binding site similarity is the most important prerequisite for a promiscuous PDB drug to bind to multiple PDB targets and that ligand flexibility has a minor impact. Molecular weight and hydrophobicity do not seem to influence whether a drug is promiscuous or not. Global structural similarity is also reflected in the pairs of similar binding sites but misses the important examples of similar binding sites in globally structurally dissimilar proteins. In particular, 15 % of all target pairs with a similar binding site are dissimilar in global structure and would have not been detected by other approaches on sequence or global structure level. As supported by their findings, protein local structural alignments bare a huge potential to infer so-far unknown drug-target relationships. Implementation in the Cloud might speed up drug development by uncovering off-targets and thus causes of adverse drug reactions early in the development pipeline. On the other Hand, predicted targets are starting points for drug repositioning.

5 Conclusions

The different biomedical applications described are designed as services for a Cloud environment. The infrastructure to support these services was chosen to fit the requirements given by the applications. Further the used Cloud middleware is solely based on open standards and capable of easy deployment of virtual machines to a local cluster or other Cloud providers. The infrastructure itself is not obliged to provide security and privacy for the private data.

By choosing and designing algorithms that preserve privacy of the input data through mechanisms like blinding, homomorphic encryption and anonymization, the desired security levels can be achieved without the necessity

to trust the Cloud provider or any other third party. For those algorithms, which could not be changed due to license restrictions, we applied a black box method and minimized the input and output information to the absolute necessary amount.

Further, as the proposed security solutions are directly targeting the developed algorithms and are therefore independent of the actual Cloud infrastructure, the services can run on virtually any Cloud platform. This allows more flexibility, interoperability and efficiency compared to platform dependent solutions. The ideas, methods and tools used to construct these privacy-preserving solutions can easily be adapted to other similar algorithms. As a result, the security risks in using public Cloud Computing upon private data decreases and new applications might arise.

6 Acknowledgments and Funding

Funding by the German Federal Ministry of Economics and Technology is kindly acknowledged as GeneCloud is part of the Trusted Cloud Initiative. Furthermore, the authors would like to thank the Center for Information Services and High Performance Computing (ZIH) at TU Dresden for generous allocations of computing time.

References

[1] Atallah, M.; Pantazopoulos, K.; Rice, J.: Secure outsourcing of scientific computations „Advances in Computers 54“, 2001.

[2] Atallah, M.; Frikken, K.: Securely outsourcing linear algebra computations „Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security - ASIACCS ’10“, New York, April 2010.

[3] Beck, M.; Kerschbaum, F.: Approximate Two-Party Privacy-Preserving String Matching with Linear Complexity „Proceedings of the 2nd IEEE International Congress on Big Data (BIGDATA)“, July 2013.

[4] Boneh, D.; Goh, E.; Nissim, K.: Evaluating 2-DNF formulas on ciphertexts „Theory of Cryptography (TCC) ’05“, 2005.

[5] Bloom, B.: Space/time trade-offs in hash coding with allowable errors „Communications of the ACM“, July 1970.

[6] Bugiel, S.; Nürnberger, S.; Sadeghi, A. *et al.*: Twin Clouds: Secure Cloud Computing with Low Latency. „12th Communications and Multimedia Security Conference (CMS’11) “, 2011.

[7] Doms, Andreas and Schroeder, Michael: GoPubMed: exploring PubMed with the Gene Ontology. „Nucleic Acids Res“, July 2005.

[8] Fortney, K.; Jurisica, I.: Integrative computational biology for cancer research. „Journal of Human Genetics“, October 2011.

[9] Gentry, C.: Fully homomorphic encryption using ideal lattices „Proceedings of the 41st annual ACM symposium on Theory of computing“, New York, 2009.

[10] Haupt, V.J.; Daminelli, S.; Schroeder, M.: Drug Promiscuity in PDB: Protein Binding Site Similarity Is Key. „PloS one“, January 2013.

[11] Kernel Based Virtual Machine, http://www.linux-kvm.org

[12] Li, C.; Wang, B.; Yang, X.: VGRAM: improving performance of approximate queries on string collections using variable-length grams „Proceedings of the 33rd international conference on Very large data bases (VLDB)“, September 2007.

[13] Open Grid Forum Working Group, „OCCI – Open Cloud Computing Interface“, 2009.

[14] Open Source Data Center Virtualization, information online: http://opennebula.org

[15] Open Stack Cloud Software, online via: http://openstack.org

[16] Roy, J.; Winter, C.; Isik, Z. *et al.*: Network information improves cancer outcome prediction. „Briefings in Bioinformatics“, December 2012.

[17] Santos, N.; Gummadi, K.; Rodrigues, R.: Towards trusted cloud computing „Proceedings of the 2009 conference on Hot topics in cloud computing“, Berkeley, 2009.

[18] Winter, C.; Kristiansen, G.; Kersting, S. *et al.*: Google goes cancer: improving outcome prediction for cancer patients by network-based ranking of marker genes. „PLOS Computational Biology“, 2012.

[19] Xen Project: http://xenproject.org

[20] Yao, A.: How to generate and exchange secrets „27th Annual Symposium on Foundations of Computer Science“, 1986.