Azerbaijan Journal of High Performance Computing, Vol 2, Issue 1, 2019, pp. 82-92
https://doi.org/10.32010/26166127.2019.2.1.82.92

*Correspondence:
Tapalina Bhattasali, St. Xavier's College (Autonomous), Kolkata, India,
tapalina@sxccal.edu

SeDReS: Secured Data Retrieval Service

Tapalina Bhattasali
St. Xavier's College (Autonomous), Kolkata, India, tapalina@sxccal.edu

Abstract

Cloud service model can provide flexible storage place to store electronic data cost-effectively. However, storage of sensitive data in the third party cloud domain may violate the privacy of data. For this reason, encryption logic needs to be applied before uploading it to cloud storage. Use of encryption logic may violate the availability of data. It is now becoming a big concern to retrieve data securely from the cloud storage. This paper considers a remote application environment, where cloud data retrieval service must be secured to preserve the privacy of sensitive data. Here multiple items are searched in a single query to make the procedure faster, and the only specific number of relevant records can be retrieved at a time to save bandwidth cost. Encryption logic is proposed in such a way that encrypted cloud data could be processed without decryption. A secured retrieval service is proposed here to ensure data availability at any time. Hybrid encryption logic is designed based on AES cipher and homomorphism properties. Limitations of fully Homomorphic encryption are avoided in proposed logic. Analytical study of AES cipher shows its efficiency to encrypt electronic data and reply vectors. Theoretical analysis of homomorphism shows its efficiency to secure searchable index, items, and query vectors required in cloud data retrieval service.

Keyword: Data Retrieval; Cloud; AES; Homomorphism; Hybrid Encryption

1. Introduction

Nowadays, users want to outsource their data and computational tasks on their data to the public cloud. It enhances flexibility, scalability, and cost-effectiveness. It is also economically feasible for the remote service provider to shift their data-centers to the public cloud. The success of data outsourcing remains in the efficiency of cloud data retrieval. Direct outsourcing of data enhances concerns regarding the privacy of sensitive information, as the data owner can not control outsourced data. Data confidentiality, along with integrity, can be achieved by encrypting the data. Availability can be achieved by effective data retrieval in real time when the reply to the query is faster and accurate. However, encryption raises the complexity by limiting the functionality of computation on outsourced data stored at third-party cloud storage. In this regard, existing techniques to retrieve data based on plain text query has no use. If all the outsourced data are decrypted before performing

any computation, it unnecessarily consumes system resource. Therefore, it is not feasible to download the outsourced data and then decrypt it locally due to the huge amount of bandwidth cost. If the decryption key is known to the cloud service provider, then the privacy of the sensitive data is compromised. As huge volumes of data are stored for large numbers of users, a single item search in a single query is time-consuming. Rank based search eliminates unnecessary network traffic by quickly sending back only the most relevant data.

The main objective of this paper is to design secured data retrieval service; so that it ensures confidentiality, integrity, and availability (CIA). Therefore, the author designs a procedural logic for outsourcing only encrypted data used in the remote application domain. Authorized users send the encrypted query to the cloud server to search electronic data of users according to their access rights. Cloud service provider searches index and returns the most relevant data. Only valid users can decrypt that data. Encryption technique must be considered in such a way that it can search for data storage without decrypting all the entries of the database. The challenge here is how to enable search and retrieval over such encrypted data [1]. Use of Homomorphic encryption is one of the feasible solutions in the cloud environment [2]. Homomorphic encryption can process cipher data without deciphering it. Fully Homomorphic encryption enhances the flexibility of processing, as any operations can be performed on cipher data. However, it has also a few limitations. The significant contribution of this paper is to determine the research gap between privacy preservation and cloud data retrieval logic. Use of only Homomorphic encryption may increase the overall complexity of the application. For this reason, hybrid encryption is considered here to provide better performance in data retrieval.

The rest of the paper is organized as follows. Section 2 presents some well-known research works on this domain. Section 3 defines the problem domain. Section 4 points out the design goal. Section 5 describes the proposed secured data retrieval service. Finally, section 6 concludes the paper.

2. Related works

Many kinds of research related to the retrieval of data from cloud storage are going on all over the world. There exist some works on Proofs of Retrievability (PoR) [3, 4] model to ensure that valid user can retrieve remotely stored data. Several variants of PoR models have been proposed, including zero-knowledge proofs of retrievability, even in the multi-client environment [5, 6, 7].

Research efforts are also directed to outsource data management along with privacy protection. There exists several searchable symmetric encryption (SSE) logic [8, 9, 10] to search and securely retrieve cipher text. However, most of these schemes support only Boolean keyword search. It considers more than one keyword in a single query to enhance the flexibility of search. Data are retrieved based on whether a searched item is present in a file or not, without considering the degree of relevance with the search query. Some of SSE logics are based on order-preserving encryption (OPE), which allow only a single search item in the query without considering privacy preservation [11]. OPE logic cannot be used in the remote healthcare domain because it leaks data privacy. There also exist a few works on multiple search items in a single

query to enable secure indexing and ranked searching [12, 13]. Ranked search enhances system usability by returning the matching files according to relevance rank (e.g., keyword frequency) [11, 12, 13]. One of the simple ranked keyword search mechanisms is based on order-preserving symmetric encryption (OPSE)[14]. OPSE may not hide similarity relevance, and robustness of the scheme becomes low. As a result, it is more prone to attacks. So server-side ranking is not secure using OPSE, as it has a high probability of leaking sensitive information. In [13], a framework is designed for topmost k data retrieval, including privacy preservation, secure indexing, and ranking based on OPE. The problem of top-k multiple search items retrieval over encrypted data is defined and solved in modified MRSE (multi-keyword ranked search over encrypted cloud data)[12] by employing coordinate matching and inner product similarity. The trapdoor can be generated in a cryptographic way to protect the query keywords. Two-round searchable encryption (TRSE)[15] supports top-k multi-keyword retrieval using vector space model and Homomorphic encryption to provide search accuracy and privacy protection.

Type of interaction between the participants is very much essential to consider the effectiveness of cloud data retrieval service [16, 17, 18, 19]. Performance complexity of most of the existing cloud data retrieval service is measured in terms of setup time, encryption time, trapdoor generation time, search time, index construction time, relevance score calculation overhead, scalability. These metrics mainly depend on number of search items, the maximum size of the searchable item set, data vector, query vector, result vector. Each of the existing cloud data retrieval service has some limitations. Multiple keyword searches are not possible for most of the cases. It is assumed in a few cases that authorized private keyword search maintains a hierarchical structure for attributes. However, it is not feasible in reality. Index construction creates overhead for a few cases.

In some cases, access patterns and additional information related to searchable items are revealed to the cloud server. Performance of data retrieval service is affected if similarity rank score of the document vector entirely depends on the document type. In some cases, trapdoor generation [20] needs more time with the increased number of attributes. If key size and encrypted trapdoor size are too large, then communication overhead increases.

Homomorphic encryption [21, 22] allows a certain amount of computation on the user's encrypted data. Attribute-based encryption [23] achieves a flexible and dynamic way to share permissions on particular parts of electronic records. In the remote application domain, the user submits an encrypted query to retrieve private data, and the cloud server performs searching of encrypted data without looking at the query in plain text format. Fully Homomorphic encryption can execute all types of operations on cipher data without decryption. Therefore, it can be said that encryption logic needs to be added to PoR models to simultaneously achieve data privacy and retrievability [24, 25, 26].

3. Problem domain

According to the literature survey, the following issues are defined under the problem domain. It is complicated to meet the performance requirement, usability,

and scalability at the same time. There still exists a trade-off between security and usability.

Cloud service providers considered as "curious" because it can analyze data including index in its storage that may violate security features. According to the information that cloud server knows, different threat models may appear with different attack capabilities [9, 10, 12], such as privacy leakage of sensitive data (cloud service provider can access outsourced data). Known ciphertext attack (cloud service provider only knows cipher text and index), known background attack (cloud service provider acquires additional knowledge including relationship among the components involved in the process to learn content of the document),statistic leakage attack (CSP could perform some statistical analysis over the search result to make estimation), association attack (CSP can analyze any association between keyword and cipher text from the index), scale analysis attack (CSP can deduce the frequency of specific keywords in the number of documents), keyword guess attack (deduce or identify certain keywords in query).

It is seen that if all data files, searchable indices, and user queries are encrypted using OPE, or other encryption algorithms using statistical information, the privacy of sensitive data may be compromised. If trapdoor generation function is deterministic, trapdoor unlinkability may not be possible. There are two significant statistic leakages [12] - term distribution (term's frequency distribution of relevance scores on each file) and inter distribution (file's frequency distribution of relevance scores on each term). They can be either extracted from cipher text or through statistical analysis over access and search pattern [14].

Definition(Curiosity). Curiosity is defined as a term that is inversely proportional to robustness. Robustness is inversely proportional to information leakage. Curiosity is proportional to information leakage. Let μ denotes the collection of all ciphers; a denotes the ciphers that should be only known to cloud server, where $\alpha \subseteq \mu$ and β denotes the set of other ciphers that must be unknown to cloud server, where $\forall \beta \subseteq \mu$, and $\alpha \cap \beta = \varnothing$. Robustness is denoted by $R=\min(\rho(\alpha)/\rho(\alpha|\beta))$ and curiosity $y=1/R$.

Although Fully Homomorphic Encryption (FHE) shows its feasibility in processing encrypted cloud data, it has some limitations. FHE using ideal lattices over a polynomial ring becomes very much complicated.

In some cases, FHE requires large storage space to store keys and cipher texts, even for the small size of raw data. Although storage space is not a problem in the cloud environment, it may increase processing time and complexity. Key management is another critical issue in a distributed environment. Use of fully Homomorphic encryption raises time complexity and computational overhead; space complexity, which may be a significant limitation for real-time applications.

4. Design goal

Exploring privacy preserving and effective search over encrypted data is very important for designing secure cloud data retrieval service. However, it is very much complicated and difficult to tackle. As users usually prefer to keep their search from being exposed to others, the essential requirement is to hide their search. Any function generation should be randomized. The cloud service provider must

not be able to deduce any relationship among the procedural logic. Any patterns used in data retrieval service must be well protected. The accuracy level of pattern matching for the original document and query should be high. Response time of data retrieval service must be faster, and the query result must be more relevant. Encryption techniques used for privacy preservation need to have low time and computational complexity and compatibility with the standard protocol used for secured data transmission in the remote framework.

Therefore, the design goal of active data retrieval service is that overall procedure must not be deterministic, and there must be less possibility of statistical information leakage by the third party. As a result, the cloud service provider could not be able to learn or predict additional information regarding data retrieval service. To activate relevant searching of outsourced data from the cloud storage, it is expected to give the following security and performance requirements.

• The query must contain multiple items and provide result based on highly relevant data to provide useful data retrieval service.

• Relevant search should ensure a fast response, low communication, and computation overhead along with privacy protection.

5. Proposed work: Secured Data Retrieval Service (SeDReS)

In this section, a secured data retrieval service (SeDReS) is proposed. A cloud hosting data service is considered in the remote application domain. SeDReS logic comprises of three types of participants in remote application area: data owner, cloud server under control of cloud service provider and data user.

Data owner needs to give valid authentication proof before uploading data. Authentic data owner records data and stores at local storage temporarily. Then, searchable indices are constructed and encrypted for the records. Original records are encrypted separately. Both encrypted data and index are packed in a digital envelope before outsourcing to the public cloud.

Cloud Server has huge storage space and computational resources to maintain a high volume of encrypted data uploaded from the remote places. It can process encrypted data without decryption. However, the cloud server cannot be fully trusted as they may misbehave under third party control. Here cloud server is assumed as semi-trusted, that is honest but curious.

Date user needs to give valid authentication proof before sending the encoded query to the cloud server. The main focus of this work is to consider data encryption technique in such a way that encrypted data (cipher) can be processed just like processing of plain data and time and computational complexity can be reduced. For this reason, hybrid encryption mechanism is considered here. SeDReS logic is based on the following notations.

• *DP - Collection of data in plain text format, denoted as a set of n data DP= {DP_1, DP_2, DP_3,...,DP_n}.*

• *DC- Collection of data in cipher text format, denoted as a set of n data DC= {DC_1, DC_2, DC_3, ... ,DC_n}.*

• *In – Searchable index associated with cipher text DC, denoted by In ={$In_h In_2$, In_3, ...,In_n}, where each sub-index $In_i$is constructed for each DP_i.*

• *In_c – Encrypted searchable index, denoted by $In_c = \{In_{c1}, In_{c2}, In_{c3},...,In_{cn}\}$.*

• *SI_R - Distinct m searchable items extracted from collection of DP, denoted as $SI_D = \{SI_{D1}, SI_{D2}, SI_{D3},.....,SI_{Dm}\}$.*

• *SI_Q – Subset of SI_R representing searchable items in query Q, denoted by $SI_Q = \{SI_{Q1}, SI_{Q2}, SI_{Q3},....,SI_{Qm}\}$.*

• *S_{SIQ}-Seed (trapdoor) works just like the one-way hash function based on searchable items SI_Q and query Q.*

• *Top_J – Topmost J (at most) relevant records are retrieved in the response of a query.*

In SeDReS, collection of data is outsourced from the data owner to cloud server in encrypted form. Before this, the searchable index is constructed and encrypted. Authorized data owner outsources both encrypted index and data. The service considers multiple items in a single query (MISQ) to search relevant data quickly along with preserving data confidentiality, integrity, and availability in a less complicated way. To retrieve data for given searchable items, authorized data user first sends authentication proof to data owner or cloud service provider, then acquires Seed, which is generated here by randomized hash function from the encrypted index, searchable items and query vector. Upon receiving Seed for a query, cloud server searches index and returns corresponding topmost J encrypted data. Cloud service providers mainly access encrypted index during the search for items. Homomorphic encryption is considered here with reduced complexity to encrypt searchable index, searchable items, and query. To make data retrieval service suitable for real-time application, sensitive data are transmitted through TLS (transport layer security) protocol. During data outsourcing, cipher data and encrypted index are packed in a digital envelope and signed by the data owner's private key and cloud service provider's public key (the identity of the entity).

A 3x3 square matrix is presented in table 1 as the participant interaction (PI) matrix to represent interaction among the participants of secured data retrieval service.

TABLE 1.3X3 participant interaction (PI) matrix

	DO	DU	CS
		Seed key	Encrypted digital envelope
DO	NULL	Secret key	containing DP and Inc,
		Decode key	Maximum value Top_J
DU	Authentication token	NULL	Encrypted multiple items, single query vectors
CS	NULL	Encrypted Top_J ranked relevant DC	NULL

All diagonal elements of PI matrix contain NULL entries. 1x2 of PI matrix represents data owner (DO) to data user (DU) interaction, 1x3 represents data owner (DO) to cloud server (CS) interaction, 2x1 represents data user (DU) to data owner (DO) interaction, 2x3 represents data user (DU) to cloud server (CS) interaction, 3x2 represents cloud server (CS) to data user (DU) interaction. 3x1 also contains a NULL value, which implies that there is no interaction from cloud user (CU) to data owner (DO), according to this interaction matrix of SeDReS logic.

5.1. Procedural Logic of SeDReS

Proposed SeDReS logic works in two phases- data outsourcing phase (DOP) and the data retrieval phase (DRP).

Definition(SeDReS). It is a service model whose lifetime LT → {DOP v DRP}. It has three member classes- {data owner (DO), data user (DU) and cloud server (CS)}, where DOP is a member of DO class and DRP is a member of both DU and CS classes. This service model has collection of eleven logic -gen_key(), construct_index(), build_vector(), enc_AES(), enc_homo(), gen_seed(), gen_query(), match_pattem(), gen_rank(), gen_reply(), dec_data(). DOP set includes {gen_key(), construct_index(), build_vector(), enc_AES(),enc_homo()} and DRP set includes *{gen_seed(), gen_query(), match_pattem(), gen_rank(), gen_reply(), dec_data()}*.

• *gen_key()* - Take security parameter as input and generate secret key K_s for Rijndael AES block cipher, Seed key K_{se} and crypto key K_c for Non-Circuit Homomorphic Encryption (NCHE). Then DO assigns K_s, K_{se}, $K_d$to authorized DU.

• *construct_index()* – DO constructs secured searchable index In_c from DP. In is encrypted into In_c. enc_homo() is called for encryption. The output is secured searchable index In_c.

• *build_vector()* – DO extracts collection of m searchable items, SI={SI_1,SI_2, ...,SI_m}, and their TF (term frequency) and IDF(inter-domain frequency) values from the collection of n records. For each record, DO constructs (m+1)-dimensional vector vecj based on tf-idf values. Searchable index In={vec_j|$1 \le i \le n$}.

• *enc_AES()* – Generate AES block cipher for data vector and reply vector. Procedural logic works on four steps- key expansion, initial round, round, final round. It is faster, secure, less prone to attacks. It provides better performance compared to most of the other symmetric ciphers. A message digest of the data record is calculated through the SHA algorithm to check integrity at receiving end.

• *Enc homoQ* – Non-Circuit Somewhat Homomorphic encryption (NCSHE) is called to encrypt searchable index, query vector, searchable items. The output is reduced cipher text with less complexity.

• *gen_seed()* – Data owner generates seed key and sends to the data user. The seed value is generated randomly from the data user query. Then message digest of this seed value is created using a one-way hash function to generate secure seed value.

• *gen_query()* – Query is represented as a vector using the vector space model, where each dimension of the vector is set to 1 or 0 according to verification that whether the specified term is queried or not.

• *match_pattern()* – Vector space model allows to compute the continuous degree of similarity between query vectors and records. According to the nearest

neighbor (NN) score, Top_J consistent record patterns are evaluated.

• *gen_rank()* – When secure Seed is received, cloud service provider computes nearest neighbor (NN) scores of data vector along with the index vector with query vector and returns encrypted result vector to the data user. The score of record on the query is derived by the inner product of the two vectors. According to the score, records are ranked.

• *gen_reply()* – According to NN score rank-id is generated for each record. Top ./records with high rank-id are sent to data user in encrypted reply vector format.

• *dec_data()* – All relevant data are downloaded and decrypted at data user side using AES and Non-Circuit Somewhat Homomorphism logic.

5.2. Hybrid Encryption Logic

In SeDReS, effective encryption mechanism is required to balance between security, privacy protection, and efficient data retrieval at remote application framework. The objective is to consider a practical approach to choose a suitable encryption logic that is more powerful and capable of reducing the gap between security and usability. Hybrid encryption logic is considered here by mainly focusing on AES block cipher and less complex Non-CircuitSomewhatHomomorphic Encryption (NCSHE).

Definition(Hybrid Encryption). Hybrid Encryption (HE) is defined as a cryptographic logic, where multiple encoding techniques are included to provide better performance with reduced space complexity, time complexity, and computational complexity. HE set includes {AES block cipher, digital envelope, public key cryptography, symmetric key cryptography, message digest, random seed calculation, homomorphic encryption}.

As vector space model and nearest neighbor (similarity measure) scoring are used for SeDReS, only addition and multiplication operations over integers are needed in Homomorphic encryption. For this reason, the complexity of Fully Homomorphic Encryption (FHE) is reduced here by considering the concept of Somewhat Homomorphic Encryption (SHE). SHE supports only addition and multiplication operations on ciphers. Each cipher includes noise component and any operation applied to ciphers increases the noise in the resulting cipher up to a threshold level. Then ciphers may not be decrypted accurately. Instead of considering as a circuit, Homomorphic function f is considered as a mathematical function to enhance the efficiency of HElib library. For this reason, NCSHE logic is considered here to enhance usability at the real-time application.

Definition(NCSHE). NCSHE is modified homomorphism where computational functions are considered as mathematical logic(without logic circuit consideration) and supports only addition and multiplication operations with the insertion of noise factors. It has four phases- initial_key(), enc(), eval(), dec() NCSHE is well suited for SeDReS in remote application domain.

According to the property of homomorphism, encryption logic passes through the following four phases.

• *Initial key():* Secret key K_5 is an odd r bit number randomly selected from the interval $[2^{r-1}, 2^r)$. Public keys are derived from identity and interval depends on bit

length of noise factor n1. Here, noise factor n2 is randomly selected and its interval depends on bit length of plain text. Key matrix is constructed using a pseudo-random set of numbers, and then converted them to modulo N. Matrix is constructed to ensure that each row or column are independent of each other; so that it cannot be predicted in any case.

• *enc():* Cipher text is represented by CT←K_5*a+2*n1+PT, where DK← K_5a+n1, a → security parameter, PT→ plain text. A random value is chosen to randomize the procedure. A matrix is constructed such that each row has only one element equal to the input, and other two equal to random numbers. Plain text or input values are solutions of linear congruences computed using Chinese Remainder Theorem. Congruences depend on the plaintext and random value.

• *eval():*-Binary addition and multiplication logic are applied to cipher text to perform the search operation. Resulting integer value κ is returned to DU.

• *dec()*: Output R'← (k mod K_s) mod n2, where R' → decipher the text, k → returned integer value, n_2→noise factor. It is a single step procedure where inverse transformation applied on cipher text matrix and then plaintext is extracted as the first element of an obtained diagonal matrix.

Here n2 ← $2^{2||PT||}$ where ||PT|| represents bit length of PT, K_s>>n1 and n1>> n2. As a result, the size of CT is reduced to 1/||PT||.

Theoretical analysis shows that the size of cipher text (space complexity), time complexity, and computational complexity are reduced, with increasing decryption accuracy.

6. Conclusion

In this paper, a secured data retrieval service SeDReS is proposed to retrieve sensitive data accurately from third-party cloud storage in real time. This logic supports CIA property of security for remote applications. According to the literature survey, it is seen that there is a trade-off between privacy and availability. In this paper, the authors try to pinpoint the research gap. SeDReS depends on the nature of the interaction among the participants. Use of vector space model and calculation of nearest neighbor score based on tf x idf weight value enhances the efficiency of data retrieval service based on the rank. This paper also focuses on the hybrid encryption technique. AES block cipher is considered to encrypt original data and reply vectors. Here, all data are considered as text-based. Therefore no image is included during encryption. Non-circuit homomorphism is used to encrypt index, searchable items, query vector.

It raises the efficiency of proposed homomorphic logic and maintains a balance between security and usability. The main aim of this research work is to implement it to serve better service to society.

References

[1] Boneh, D., Di Crescenzo, G., Ostrovsky, R., & Persiano, G. (2004, May). Public key encryption with keyword search. In *International conference on the theory and applications of cryptographic techniques* (pp. 506-522). Springer, Berlin, Heidelberg.

[2] Jain, N., Pal, S. K., & Upadhyay, D. K. (2012). *Implementation and analysis of*

homomorphic encryption schemes. International Journal on Cryptography and Information Security (IJCIS), 2(2), 6.

[3] Bowers, K. D., Juels, A., & Oprea, A. (2009, November). Proofs of retrievability: Theory and implementation. In *Proceedings of the 2009 ACM workshop on Cloud computing security* (pp. 43-54). ACM.

[4] Li, J., Tan, X., Chen, X., & Wong, D. S. (2013, September). An efficient proof of retrievability with public auditing in cloud computing. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems* (pp. 93-98). IEEE.

[5] Zhu, Y., Wang, H., Hu, Z., Ahn, G. J., Hu, H., & Yau, S. S. (2011, March). Dynamic audit services for integrity verification of outsourced storages in clouds. In *Proceedings of the 2011 ACM Symposium on Applied Computing* (pp. 1550-1557). ACM.

[6] Abo-Alian, A., Badr, N. L., & Tolba, M. F. (2015). Auditing-as-a-service for cloud storage. In *Intelligent Systems' 2014* (pp. 559-568). Springer, Cham.

[7] Curtmola, R., Garay, J., Kamara, S., & Ostrovsky, R. (2011). Searchable symmetric encryption: improved definitions and efficient constructions. *Journal of Computer Security, 19*(5), 895-934.

[8] Wang, C., Cao, N., Li, J., Ren, K., & Lou, W. (2010, June). Secure ranked keyword search over encrypted cloud data. In *2010 IEEE 30th International Conference on Distributed Computing Systems*(pp. 253-262). IEEE.

[9] Cao, N., Wang, C., Li, M., Ren, K., & Lou, W. (2013). Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on parallel and distributed systems, 25*(1), 222-233.

[10] Boldyreva, A., Chenette, N., & O'Neill, A. (2011, August). Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *Annual Cryptology Conference* (pp. 578-595). Springer, Berlin, Heidelberg.

[11] Yu, J., Lu, P., Zhu, Y., Xue, G., & Li, M. (2013). Toward secure multikeyword top-k retrieval over encrypted cloud data. *IEEE transactions on dependable and secure computing, 10*(4), 239-250.

[12] Martınez, S., Miret, J. M., Tomas, R., & Valls, M. (2013). Security analysis of order preserving symmetric cryptography. *Applied Mathematics & Information Sciences (AMIS), 7*(4), 1285-1295.

[13] Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y. T., & Li, H. (2013, May). Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security* (pp. 71-82). ACM.

[14] Hu, H., Xu, J., Ren, C., & Choi, B. (2011, April). Processing private queries over untrusted data cloud through privacy homomorphism. In *2011 IEEE 27th International Conference on Data Engineering*(pp. 601-612). IEEE.

[15] Li, M., Yu, S., Cao, N., & Lou, W. (2011, June). Authorized private keyword search over encrypted data in cloud computing. In *2011 31st International Conference on Distributed Computing Systems*(pp. 383-392). IEEE.

[16] Wang, C., Cao, N., Ren, K., & Lou, W. (2011). Enabling secure and efficient ranked keyword search over outsourced cloud data. *IEEE Transactions on parallel and distributed systems, 23*(8), 1467-1479.

[17] Li, M., Yu, S., Ren, K., Lou, W., & Hou, Y. T. (2013). Toward privacy-assured

and searchable cloud data storage services. *IEEE Network, 27*(4), 56-62.

[18] Zhao, Y., Chen, X., Ma, H., Tang, Q., & Zhu, H. (2012). A New Trapdoor-indistinguishable Public Key Encryption with Keyword Search. *JoWUA, 3*(1/2), 72-81.

[19] Smart, N. P., & Vercauteren, F. (2010, May). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography* (pp. 420-443). Springer, Berlin, Heidelberg.

[20] López-Alt, A., Tromer, E., & Vaikuntanathan, V. (2012, May). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing* (pp. 1219-1234). ACM.

[21] Benaloh, J., Chase, M., Horvitz, E., & Lauter, K. (2009, November). Patient controlled encryption: ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM workshop on Cloud computing security* (pp. 103-114). ACM.

[22] Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security* (pp. 89-98). Acm.

[23] Swaminathan, A., Mao, Y., Su, G. M., Gou, H., Varna, A. L., He, S., ... & Oard, D. W. (2007, October). Confidentiality-preserving rank-ordered search. In *Proceedings of the 2007 ACM workshop on Storage security and survivability* (pp. 7-12). ACM.

[24] Baek, J., Safavi-Naini, R., & Susilo, W. (2008, June). Public key encryption with keyword search revisited. In *International conference on Computational Science and Its Applications* (pp. 1249-1259). Springer, Berlin, Heidelberg.

[25] Sun, W., Yu, S., Lou, W., Hou, Y. T., & Li, H. (2014, April). Protecting your right: Attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*(pp. 226-234). IEEE.

Submitted 17.03.2019
Accepted 19.05.2019