A Novel Safe File Sharing Method Based on Cloud Storage Structure in DHT Networks

Zhongning Lu¹ and Gaofeng Shen²

1. College of soft, ZhengZhou University of Light Industry, Zhengzhou 450002, China
2. School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou Henan 450002, China
zhnily@163.com

Abstract

Safe file sharing mechanism in cloud storage is one of the most significant elements that affects the development of security technology. With application of homomorphic key agreement mechanism, Shamir secret sharing algorithm, Bloom Filter and B-tree search algorithm and Rsync data updating algorithm, a safe file sharing scheme in cloud storage is established based on cloud storage infrastructure of DHT networks. Under the circumstance of cloud storage, the scheme can make full use of the sharing mechanism and achieve the confidentiality and integrity of the file. Based on DHT networks, the analysis and test of data sharing are conducted in the paper. Together with file upload and download, the test of data encryption and decryption is done on the basis of Dropbox. As the result shows, the scheme puts forward a safe and effective solution to safe file sharing in cloud storage.

Keywords: Cloud storage, DHT networks, file sharing, security technology

1. Introduction

Cloud storage is a new extension of the concept and developed on the basis of cloud computing [1-3]. Cloud storage is mainly provide online storage services based on the Internet to the user, is a set of transparent access to the resource pool resource management and access to the Internet by a large number of storage resources and storage resources of these organizations to provide users with control technology consisting of [4,5].

With the development of cloud storage technology and applications, secure access to cloud storage system of private data, data confidentiality, privacy protection and file sharing security issues become increasingly prominent. Especially in the public cloud storage system, how to ensure the confidentiality of sensitive data and privacy, and provide the necessary mechanisms for sharing data security is becoming a hot topic of the current cloud storage security research. Current cloud storage data sharing security threats mainly from the following three aspects: (1) confidentiality, ISP intentionally or misuse of information leakage resulting data; (2) the integrity of the data from unauthorized entities modify, insert, delete etc; (3) cloud storage system in an organization's server cluster failure or loss of data that users cannot access the data.

Based on DHT network, we propose a scheme for data security shared cloud storage system, the program can not only provide confidentiality, integrity protection of shared data, but also facilitates the owner or authorized user of the data off-site processing of data, for more Multi-authorized users access to shared data, save storage costs.

2. Storage Cloud Security Related Work

In the cloud storage mode, data is stored in a third-party cloud storage platform, from the data owner (Owner, O) control, which depends on the ISP safe level (Service Provider, SP). Traditional data privacy protection method is to encrypt user data stored in the cloud server, such as the literature [4-5] and so on. Under these conditions, once the server-side leak or compromise, it could easily lead to data integrity and user privacy disclosure threat. Therefore, if the SP cannot be trusted, the user needs to consider the confidentiality, integrity and privacy protection mechanisms for cloud data storage, which has become a hot topic of research in recent years, cloud storage security and privacy protection [5-6].

[7] proposed by the data key to encrypt the data, and then use the control key to encrypt data key. Control is the key in this scheme by the key management of third-party managers, there are credible and lead to key managers control key compromise security risks. After the literature [8] Vanish system through the key threshold cryptography processing, key slice distributed by Vanish system directly into DHT network. Thus, an attacker can sniff attack [7] or jump attack [9] get enough key slice reconstructed key. [10] on the Shamir secret sharing algorithm [11] has been improved by extending the length of the key parts of the system to resist Vanish exist jump attack, using the RSA public key encryption to protect against sniffing attacks. The program can better solve the literature [7-8] safety defect exists, but in terms of efficiency and user privacy is still insufficient.

When large amounts of data stored in the cloud server, how to achieve efficient retrieval of encrypted data has become an urgent problem to be solved. Song et al proposed a searchable first encrypted security model [12], uses a two-story structure of the encrypted file keywords encrypted. Subsequently, Goh et al proposed an effective safety index model Z-IDX [13], the model uses a pseudo-random function, and Bloom filters (Bloom Filter, BF). Furthermore, Curmola [14] and Chang [15], who also use a similar method index, in this method, each file an encrypted hash table index, an index table for each keyword and consists of an encrypted file that contains the keyword constitute a set of identifiers. Bloom filter method by criteria established for each file hash index, and in the cloud server, the method can effectively reduce the index terms of storage space to store them.

Based on DHT network, proposes a scheme for data security shared cloud storage system, the program combines homomorphic key negotiation mechanism, Shamir secret sharing algorithm, Bloom Filter combined with the B + tree search algorithm, Rsync data update algorithm provides User privacy protection scheme based on data sharing under a private cloud storage conditions. Its main advantages are as follows: First, the key into the control key and the data key, key to encrypt data using the data by controlling key encryption key data. By simplifying the classification key key management, enhanced security keys; two, using RSA key negotiation with the state to generate the control key, the information is complete interactive ciphertext, avoiding untrustworthy due to user privacy SP disclosure issues; Third, the use of secret sharing method for data ciphertext and key ciphertext processing, multiple ciphertext fragment sent to the DHT network, avoiding the fragmentation caused by damage to or loss of data unrecoverable problems. Fourth, by finding algorithm BF and B + combining the use of BF algorithm is efficient space utilization and B + tree quickly pinpoint characteristics, can achieve the data for efficient storage access, and quickly and accurately find; five, using Rsync synchronization update algorithm, DHT network node dynamically updated data, we propose a DHT node data effectively extend the validity of the new mechanism to effectively prevent the data due to DHT network jitter and loss.

3. Related Infrastructure and System Model

The next section focuses on a private cloud storage conditions required for the proposed construction of relevant basic concepts and algorithms based on user privacy protection scheme for data sharing, including system models, descriptions and safety related algorithms assume three parts.

3.1 System Model

Based on DHT (Distributed Hash Table) network, using homomorphic key negotiation, BF combination with B + tree search algorithm and Rsync update algorithm, established a system suitable for cloud storage data security sharing scheme, the system model of the program are as follows Figure 1. First, the data owner O data and data encryption keys, and by Shamir secret sharing algorithm ciphertext fragment processing their information, then slice information is uploaded to DHT network. Authorized users (User, U) when the data information query through the combination of BF and B + tree lookup to obtain information and decrypt ciphertext obtained data. Due to the presence of DHT network jitter, data storage has storage time limits, we use Rsync algorithm for data update operation, thereby extending the validity of DHT node data. Which, using the method of dispersing DHT network data storage, either to improve the utilization of resources, but also can reduce the impact of a server cluster failover data integrity.

Figure 1. A System Model

3.2. Relevant Basic Algorithm

Related basic algorithm of this section focuses on program design required, including consultation with key state and Shamir secret sharing algorithm, Bloom Filter combination with B + tree search algorithm, the algorithm of three parts Rsync data updates.

3.2.1. RSA Keys in Consultation with the State: Secret homomorphic (Private Homomorphic) thought by Rivest *et al.* [16] proposed. Users using the same state properties, it is possible without using the decryption key conditions ciphertext direct operation, so as to realize the corresponding plaintext operation, the operation will not disclose information about the plaintext, these basic operations, such as: addition, subtraction, multiplication and division. Literature [17, 18] has been

demonstrated, RSA has a multiplicative homomorphic encryption scheme, and its multiplicative homomorphic properties are summarized as follows:

For any two plaintext m_1, m_2 , can be obtained $E(m_1) = m_1e$, $E(m_2) = m_2e$, there are $E(m_1) \circ \times E(m_2) = (m_1e) \circ \times (m_2e) = (m_1 \times m_2) e = E(m_1 \times m_2)$, namely RSA cryptosystem meet multiplicative homomorphic sex.

Typically, in order to ensure the confidentiality of the data stored in the cloud, the data owner O requires data key $DK = (dk_1, dk_2, \dots, dk_n)$ encrypted data $M = (m_1, m_2, \dots, m_n)$, and Key data stored in the server. This protection mechanism exists two problems: one, if the n-th component DK sent directly to the DHT network vulnerable to sniffing attacks and jumping attacks, reduce the security of the system; two, if the service provider SP cannot be trusted, or is the key transfer process between the primary and the user U O being intercepted or stolen, cloud data security will lose protection.

In this paper, based on RSA key negotiation with the state [17] to generate the control key CK, CK encrypted using the control key DK, and the use of Shamir secret sharing algorithm for DK to be fragmented, and finally sends the content to the DHT network fragmentation. Only the user U gets control key can decrypt the data. Data during transmission remains ciphertext. Therefore, this method can not only protect the confidentiality of the information exchange process, but also to avoid the SP direct access to the user's private information.

3.2.2. BF Combined with the B + Tree Search Algorithm: The system uses the BF and B + trees are combined to achieve efficient space utilization and fast and accurate search.

BF algorithm, data of a set of n data blocks $M = (m_1, m_2, \dots, m_n)$ by the k hash functions $h_1, h_2, \dots, h_k$ is mapped to the bit string length of m vector V, each hash function and the function is independent of the range $\{0, 1, 2, \dots, m-1\}$. When using the BF algorithm, each assigned a probability of making someone is: $1/m$, the probability of 0 to $1-1/m$, this process execution kn Communist times assignment process, at the end of the algorithm, someone still 0 probability:

(1)P is the false positive rate:

(2)BF complete collection is stored, you only need to save an average of m/n bits for each element, space complexity is $O(m)$. With Bloom Filter preliminary filtration, the remaining np a B + tree algorithm using a second look. Its time complexity is $O(k) + O(\log np)$.

In this paper, BF and B + tree search algorithm, mainly in the following two advantages: (1) BF algorithm for computing simple and easily implemented in hardware; (2) BF algorithm for an exact match (B + tree) pre-processing operation, which uses BF algorithm preliminary filter, filter out the vast majority of users do not meet the needs of data items, and then use B + tree is an exact match, accelerate query efficiency (Find item reduction) B + tree; (3) if only using the B + tree, for n data information need only store the n nodes, plus pointer information to be added, its spatial complexity is greater than $O(n)$.

3.2.3. Rsync Algorithm Update: DHT network data can be viewed as a distributed storage network utilizes a distributed hash table methods. DHT network using the periodic auto-update mechanism, that is, each after a specific period of time, it will discard old data to make room to store new data. This disturbance characteristic data provide convenient conditions for self-destruction, but there are also disadvantages to save the time limit cannot be customized. To prevent sensitive data because of periodic updates DHT lost, the system uses the Rsync algorithm update cycle in DHT network nodes to the data re-fill, creating a self-updating mechanism to define data storage time limit.

1996, Tridgell and other proposed a fast and robust synchronization method Rsync algorithm [19]. Algorithm mainly through two hash algorithm: alder-32 and MD5, use alder-32 calculation speed and collision rate characteristics, combined with one-way MD5 good advantage, improve the matching speed and reduce the amount of calculation. The basic idea is: SP-side computing two hash values for each data block, and then sends it to the O side. O terminal block to find a match in the data block from the hash value received, and then send the results to the SP-side match. SP end result is generated using the data in the update.

3.3. Model Description and Security Assumptions

This section describes the program focused on building solutions required for the model and assumptions related to security.

3.3.1. Program Model Description: In this system, there are three entities: the data owner $O = \langle IDO, IPO \rangle$, authorized users $U = \langle IDU, IPU \rangle$, ISP $SP = \langle IDSP, IPSP \rangle$. Which represents the identity information ID, IP represents the network location.

Model consists of the following seven steps:

(1) Data key generation: $DKeyGen(\lambda, n) \rightarrow DK$. λ for system security parameter, n is the number of data blocks. Algorithm generated based on λ and n a data key $DK = (dk1, dk2, \dots, dkn)$.

(2) the encrypted data: $Enc(DK, M) \rightarrow C$. Through key $DK = (dk1, dk2, \dots, dkn)$ encrypts plaintext data $M = (m1, m2, \dots, mn)$ to give the ciphertext data $C = (c1, c2, \dots, cn)$, i.e., $ci = Enc(dki, mi)$, wherein $(1 \leq i \leq n)$.

(3) in consultation with the State Key: The RSA cryptosystem. O SP utilizing the generated random number and its own secret random number to calculate the control key CK. U SP and O obtained random number, using the RSA multiplication homomorphism calculate CK without leaking additional information.

(4) Data-key encryption and fragmentation: $S[Enc(CK, DK)] \rightarrow CDK$. First with CK encryption DK, then again fragmented, too $CDK = (Cdk1, Cdk2, \dots, Cdkn)$.

(5) Data Key Recovery: $RDKey(t, CK, CDK) \rightarrow DK$. To obtain a sufficient key fragments from the network, reconstructed CDK, CK decrypted and then use to obtain the data key DK.

(6) to decrypt the ciphertext data: $Dec(DK, C) \rightarrow M$. Through key $DK = (dk1, dk2, \dots, dkn)$ decrypt the ciphertext data $C = (c1, c2, \dots, cn)$, to obtain the plaintext data $M = (m1, m2, \dots, mn)$, i.e., $mi = Dec(dki, ci)$;

(7) Rsync algorithm based on data updated. By comparing the hash alder-32 value $ha(i)$ and MD5 scattered in $hm(i)$, where $(i = 1, 2, \dots, n)$. Update hash value different block, if there is no actual data update, the refresh its storage duration.

3.3.2. Safety Hypothesis: For the above model, we have the following security assumptions:

(1) SP is not credible that the SP may be profitable for some purposes, data analysis stored thereon, and leaked to other non-authorized users.

(2) U is credible, U will not take the initiative to disclose its plaintext data acquisition, control key, the data key and ciphertext data, but U may be affected by uncontrollable factors or misuse, etc. lead to information disclosure.

(3) monitoring the attacker

Aggressive behavior (4) the attacker's behavior is usually an afterthought. Attackers typically not aggressive real-time access what data value only after the data is used to determine whether the attacker will attack to get the data.

4. Secure Cloud Storage File Sharing Program

This section first gives a detailed description of the system model in the algorithm and then introduce cloud storage data security sharing scheme based on DHT network.

4.1. Design

(1) $DKeyGen(\lambda, n) \rightarrow DK$

Depending on the security parameter λ and system data blocks divided by the number of M n , automatically generated data key $DK = (dk1, dk2, \dots, dkn)$, used to encrypt the plaintext data blocks.

(2) $Enc(DK, M) \rightarrow C$

Use symmetric encryption algorithm AES, for each block of plaintext data with 256 m_i (other lengths cannot be used? If other lengths can be used, where the 256-bit required) dki encryption key, to get the ciphertext block c_i .

(3) homomorphic key agreement

O will be sent after the public key encryption and time stamp their own identity information with the SP to SP : $E_{pksp}(IDO) = (IDO || timeO) \text{ pksp} \rightarrow SP$; SP O sent a message received and decrypted, get IDO and $timeO$, then generate random data RSP , SP will get the IDO and $timeO$ and their identity information, time stamps, public key encryption O 's back O : $E_{pkO}(RSP) = (IDO || timeO + IDSP || RSP || timeSP) \text{ pksp} \rightarrow O$; O decryption get RSP , and then generates a random number RO , calculate $CK = RORSP$; U will own identity and public key encryption timestamp respectively after SP and O and O were sent to the SP :

$E_{pksp}(IDU) = (IDU || timeU) \text{ pksp} \rightarrow SP$

$E_{pkO}(IDU) = (IDU || timeU) \text{ pkO} \rightarrow O$

After receiving the request message SP and O were returned by pkU encrypted random number that $ROpkU$ and $RSPpkU$; multiplication using the RSA encryption system with normality, U computing $ROpkURSPpkU = (RORSP) \text{ pkU}$, then you can get the decryption $CK = RORSP$.

(4) $S[Enc(CK, DK)] \rightarrow CDK$

The $DK = (dk1, dk2, \dots, dkn)$ of the n -th block in series, and by using homomorphic encryption key CK consultations have been CDK . Then CDK Shamir secret sharing algorithm is divided into n fragments, namely $CDK = (Cdk1, Cdk2, \dots, Cdkn)$, and upload it to DHT network.

(5) $RDKey(t, CK, CDK) \rightarrow DK$

By pinpoint BF filter and $B +$ tree, who? Can be obtained from a sufficient number of key network fragmentation, and reconstruct the CDK , and then produced by CK decryption key negotiation CDK , obtain $DK = (dk1, dk2, \dots, dkn)$.

(6) $Dec(DK, C) \rightarrow M$

Symmetric encryption algorithm AES, for each ciphertext block c_i , by 256 key dki decrypted plaintext block m_i .

(7) Rsync algorithm data update

To ensure data integrity on the cloud server, the system uses the Rsync algorithm update, stored on cloud servers and data, including data ciphertext key ciphertext two parts. Following an example to the ciphertext data update method.

SP calculations ciphertext $C = (c1, c2, \dots, cn)$ values for each data block hasp (i) and hmisp (i), and will send it to O; O received SP sent hasp (i) and hmisp (i) and the establishment of the hash table (hasp (i), hmisp (i)) sp; O own computing $C = (c1, c2, \dots, cn)$ of hao each data block (i) and hmo (i), and the establishment of another hash table (hao (i), hmo (i)) o; compare two hash tables, if two elements are equal to the same position, that is a match, the refresh time of the data stored on the SP; if equal, O sends the data to the corresponding location on the SP, SP replace the data with its original location, a data update operation is completed.

As mentioned above, in a private cloud storage conditions proposed scheme based on user privacy protection in data sharing, data sharing process can be divided into stages, and data encryption and decryption phase two parts.

In the encryption phase, O generated by DKeyGen data key, the encrypted data block with Enc, while with S [Enc (CK, DK)] key encryption and data fragmentation processing. After the ciphertext data and key ciphertext C CDK uploaded to DHT network.

In the data decryption phase, U first searches the network for more than enough ciphertext fragment, then a request message to the O and SP respectively, by accepting the returned information to calculate CK, use RDKey and Dec were key to decrypt the data and plaintext data, complete the data sharing.

5. System Test

The following experiments, analyzed to verify the above data security sharing scheme based on DHT network, mainly to test the file upload / download speed, Shamir fragmentation, data encryption and decryption time and multiplicative homomorphic sex-based RSA get the key test.

5.1. Experimental Environment

In this paper, the experimental environment for: Intel Core 3.20GHz, 8GB RAM, Win7 32-bit operating system. Upload network disk using Dropbox, network access bandwidth 100Mbps. Encryption algorithm AES encryption library using AesLib key length of 256 bits.

5.2. Experimental Results and Analysis

(1) file upload and download time

Figure 2, Figure 3 shows the different size of the data file uploading and downloading the average used time, it can be seen from the figure, the data upload and download files and the file size is approximately linear. No congestion in the network, upload and download data files 1G size were used during 1828 seconds and 7062 seconds.

Figure 2. Data File Upload Efficiency

Figure 3. Data file Download Efficiency

(1) Shamir slice consuming test

2KB of an encrypted file, the threshold rate remained at 90%, in number of different pieces of key, the measured data shown in Figure 4:

Figure 4. Shamir Slice CoTsuming test

(3) data encryption and decryption test

Figure 5. Data Encryption

Figure 6. Data Decryption

(4) to obtain the key test is based on RSA multiplicative homomorphic sex

Using RSA multiplicative homomorphic sex, U just a multiplication operation and a second decryption operation, you can get $CK = RORSP$; if you do not use RSA multiplicative homomorphic sex, U need to decrypt $ROpkU$ and $RSPpkU$ do multiplication operations, namely need twice decryption operation and a multiplication operation. That is twice the decryption respectively RO and RSP, then multiplication get $CK = RORSP$. Because RSA decryption operation is relatively time-consuming, so the idea of the proposed acquisition to enhance the efficiency of the decryption key. RO uses 512bit random number and a random number RSP 512bit in RSA modulus under different values of n ($n = 256, 512, 1024, 2048, 4096$) cases, respectively, and do not use multiplication multiplicative homomorphic homomorphic two kinds of the RSA algorithm, the two kinds of comparison tests key acquisition method has been to obtain a key mechanism in case of two kinds of CK generation time comparison results, shown in Figure 7.

Figure 7. RSA Multiplicative Homomorphic Key Acquisition

6. Security Analysis

In this paper, under the assumption that their safety will be fully guaranteed security, starting with the confidentiality and integrity of both explained.

1. Confidentiality

This article is distributed to DHT network data is through the control key encryption key, and after its ciphertext Shamir secret sharing algorithm re-sliced. After two heavy encryption can effectively resist sniffing attacks and jumping attacks. Using the aforementioned formulas expressed in a key step program, and then assume a rival, in the above encryption, come to attack the results invalid. For SP credible, this paper based on the RSA key agreement with the state in order to generate the control key mechanism, based on the following BAN logic [20] to give proof of their safety.

The main purpose of this paper is to establish a mechanism to control key CK through tripartite session, the initial assumptions of the agreement are:

- (2) $SP \models \rightarrow O$, i.e. the public key of SP believe O PKO;
- (3) $O \models \rightarrow SP$, i.e., SP's public key is O believe PKSP;
- (4) $O, SP \models \rightarrow U$, i.e. O, SP believes is the public key of U PKU;

Step tripartite exchange agreement can be formalized as follows:

- (1) $SP \triangleleft \{IDO \parallel timeO\}$
- (2) $O \triangleleft \{IDSP \parallel timeSP \parallel RSP + IDO \parallel timeO\}$
- (3) $O \triangleleft \{RO\}$, $O \triangleleft \{RSP\}$, seen by BAN logic rules, O can be constructed $CK = RORSP$
- (4) $SP \triangleleft \{IDU \parallel timeU\}$, $O \triangleleft \{IDU \parallel timeU\}$
- (5) $U \triangleleft \{RSP\}$, $U \triangleleft \{RO\}$, by BAN logic rules, O computable $ROpkURSPpkU = (RORSP) pkU$, then decrypt $CK = RORSP$

Above steps (1) and step (2) to achieve mutual authentication can prevent middle attack; transmission of random numbers are used other public-key encryption, only the recipient's private key can decrypt legitimate to protect the security of random numbers; interactive information added timestamp can prevent replay attacks. This gives a homomorphic RSA key negotiation mechanism proposed is safe and efficient.

2, integrity

Cloud storage data environment to meet the confidentiality of the system, unauthorized users cannot get the data information, and therefore cannot modify the data, insert, delete and other damage data integrity operations. Cloud storage system server clusters appear short or permanently any component failures are possible. The system uses the Shamir secret sharing algorithm, simply get a slice of n t a system can be reconstructed from data loss or damage to individual slice does not affect the integrity of the data. By Rsync update algorithm, making DHT network to maintain real-time updates of data, and by effectively extending the deadline DHT node data storage mechanism, it does not because DHT network with state properties lost, ensuring da

7. Conclusion

Based on DHT network, using RSA key agreement with the state, BF combination with B + tree search algorithm, Rsync data update algorithm, proposed a cloud storage system for secure file sharing solution, to enhance the security of cloud data storage environments, ensure their privacy, integrity. Among them, the homomorphic key agreement allows authorized users get more control key and in order to achieve the purpose of file sharing. SP in the key negotiation process completely unable to get control of the key information to ensure the privacy of user data. When using DHT network and other errors Shamir secret sharing algorithms such loss or malfunction occurs in the network, you only need to re-select other nodes to obtain data, without the use of retransmission mechanism to improve the resource utilization.

References

- [1] J. Wu, P. Wyckoff and D. K. Panda, "PVFS over InfiniBand: Design and Performance Evaluation", In: Proceedings of 2003 International Conference on Parallel Processing, (2003), pp. 107-115.
- [2] S. A. Weil, K. T. Pollack and S. A. Brandt, "Dynamic Metadata Management for Petabyte-Scale File Systems", In: Proceedings of the ACM/IEEE SuperComputing Conference, (2004), pp. 35-47.
- [3] D. Ellard, J. Ledlie and P. Malkani, "Passive NFS Tracing of Email and Research Workloads", In: Proceedings of the Second USENIX Conference on File and Storage Technologies (FAST'03), San Francisco, CA, (2003) March, pp. 203-216.
- [4] C. Wang, Q. Wang and K. Ren, "Privacy-preserving public auditing for data storage security in cloud computing", In: INFOCOM, 2010 Proceedings IEEE, IEEE, (2010), pp. 1-9.
- [5] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing", In: Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on. IEEE, vol. 1, (2012), pp. 647-651.
- [6] C. Wang, S. S. M. Chow and Q. Wang, "Privacy- preserving public auditing for secure cloud storage", Computers, IEEE Transactions on, vol. 62, no. 2, (2013), pp. 362-375.
- [7] Y. Tang, Patrick P. C. Lee and J. C. S. Lui, "FADE: Secure overlay cloud storage with file assured deletion [A]", In: Proc of the SecureComm'10. New York: ACM Press, (2010), pp. 380-397.
- [8] R. Geambasu, T. Kohno and A. A. Levy, "Vanish: Increasing data privacy with self- destructing data", In: Proc of USENIX Security. 09. Berkeley, CA: USENIX Association, (2009), pp. 299- 316.
- [9] S. Wolchok, O. S. Hofmann and N. Heninger, "Defeating vanish w ith low- cost sybil attacks against large DHTs [A]", In: Proc of NDSS 2010. Washington, DC: Internet Society, (2010), pp. 37- 51.
- [10] L. Zeng, Z. Shi and S. Xu, "Safevanish: An improved data self-destruction for protecting data privacy", In: Proc of CloudCom 2010. New York: IEEE Press, (2010), pp. 521- 528.
- [11] A. Shamir, "How to share a secret", Communications of the ACM, vol. 22, no. 11, (1979), pp. 612- 613.
- [12] D X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data", In: Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, (2000), pp. 44-55.
- [13] E J Goh, "Secure Indexes", IACR Cryptology ePrint Archive, 2003, (2003), p. 216.
- [14] R. Curtmola, J. Garay and S. Kamara, "Searchable symmetric encryption: improved definitions and efficient constructions", In: Proceedings of the 13th ACM conference on Computer and communications security. ACM, (2006), pp. 79-88.
- [15] Y C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data", In: Applied Cryptography and Network Security. Springer Berlin Heidelberg, (2005), pp. 442-455.
- [16] R. Rivest, A. Shamir and M. Dertouzos, "On data banks and pivacy homomorphisms", Foundations of Secure Computation, vol. 7, no. 1, (1978), pp. 169-177.
- [17] M. Tebaa, S. El Hajji and A. El Ghazi, "Homomorphic encryption applied to the cloud computing security", In: Proceedings of the World Congress on Engineering, vol. 1, (2012), pp. 4-6.

- [18] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE", *SIAM Journal on Computing*, vol. 43, no. 2, **(2014)**, pp. 831-871.
- [19] A. Tridgell and P. Mackerras, "The sync algorithm", **(1996)**.
- [20] M. Burrows, M. Abadi and R. Needham, "A Logic of Authentication. Digital Systems Research Center", Technical Report: vol. 39, **(1989)**.