Received: December 13, 2017

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Proficient Hamming Weight based RSA-MD5 Security for Data Storage in Multi Cloud Environment

Aruna Guruvaya Mogarala[1*] **Mohan Govinda Sa Kabadi[2]**

[1]*Sai Vidya Institute of Technology, Computer Science and Engineering, India*
[2]*Presidency University, Computer Science and Engineering, India*
* Corresponding author's Email: arunaphd2017@gmail.com

Abstract: Cloud computing method provides the method for the Storing, managing and processing of the data through the internet. The cloud provider manages the data and resources and user can outsource the data in the cloud. The user can process the data with light hardware machines and provider resources can be used to process the data. The user has lot of advantageous like access the data at anywhere, backup the data etc., in using the cloud system. This helps in improving the cloud system as the number of users is increasing rapidly in the cloud. The increasing use of the cloud requires less computational time and more secure system. Many encryption method is used in the cloud to improve the security but most of the methods requires higher processing time. In this research, hamming weight with the Rivest–Shamir–Adleman (RSA) system is used to encrypt the data with limited computational time. The encryption and decryption time of the RSA is decreased by using the less complex value. The experimental results of the proposed system are compared with six other methods and it is evident that the proposed mechanism outperforms the other methods in success rate. The success rate improved up to 99.94% and other deadlock also reduced. This also have the higher Transaction per Second when compared to other existing systems. The proposed method generates the key using simple algorithm and this helps in limited use of resources, hence increases the performance of the system.

Keywords: Cloud computing, Encryption, Hamming weight, Limited computational time, Rivest-Shamir-Adleman.

1. Introduction

The cloud computing is developing rapidly due to their features of storing, managing and processing of the large amount of data through the internet [1]. The Online Learning websites takes advantage of the cloud computing and to store and process the large amount of data [2]. Many companies such as Google, Amazon, e-commence companies like e-bay and social media like Facebook, twitter etc. provide cloud services [3]. The data owner can outsource their data in the cloud and that data may contain sensitive information about the individual and provider have to secure the data. The cloud computing provides the significant economic benefits to the data owners by outsourcing the data in the cloud [4]. The development of the cloud helps the user to store the data and also provide the security for their data. More private enterprise stores their database in the cloud and this increases the demand for the security and availability of their data [5]. But current method used in the cloud doesn't provide both the features of the cloud. This causes the rise of threats for the private data of entrepreneur and it is valued for the competitor of the entrepreneur. The various attack has been used by the attacker in the cloud to obtain the data.

For example, Denial of Service (DoS) is the type of the attack that disclose the private data and causing interruption for the user [6]. Many research pointed out the flaws in the cloud such as the transient and permanent failure. In the transient failure, the data can be recover from the backups made by the provider and it protects the integrity of the data [7]. The powerful outage like DoS attack and internal error in the data centre can also cause the transient failure [8]. Most cloud service provides

the availability of nearly 99.95% in a year. But the security of the cloud is need to be increased [9]. The encryption and decryption computation time in the cloud security is still a major issue. In this method, hamming based RSA (HRSA) method is used to encrypt the data in the TPC-H benchmark dataset and MD5 algorithm is used to analysis the data integrity of the function. RSA is one of public key cryptography system and it is used to secure the data transmission. The encryption key is public key and key used for the decryption is different from the encryption key, which is kept as secrete. MD5 is the hash value with 128-bit value and it is used to analysis the data integrity of the encryption system construct with RSA [10]. This proposed method increases the speed of the encryption and decryption process and reduces the computational overheads of the RSA. This requires minimum resources for security development and remaining resource is used for the other computational purpose and due to this performance of the system is increased. The result of the proposed system is comparing with six different existing systems and it shows that the proposed method outperforms other existing systems.

The Paper is divided into four sections with the explanation of the proposed system and their respective experimental result. In the section I, consists of the literature review of some latest papers and section II consist of the description of the proposed system. In section IV, the experimental result with other system is shown.

2. Literature review

In this section, some of the latest clouds computing encryption method are discussed with the process. The useful of the paper and their limitation are also mentioned in this section.

C. Huang, R. Lu, and K.K.R. Choo [11] proposed the method for the secure and flexible cloud assistance associative rule mining in the horizontally partitioned dataset. This method helps the user to upload their data in the cloud and help to mine the data without minimum risk of the privacy leakage. They also showed that the technique achieves the privacy and also shows the resistance against the collusion attack. This method has the low computation cost compare to the other existing method. This method only works on the horizontal dataset.

M. Ahmadian, F. Plochan, Z. Roessler, and D.C. Marinescu [12] used secure proxy to carry out the transformation and there was no change made on the cloud. This technique can be applicable to all the NoSQL data model and in this research, it was applied to the Document-store data model. This method was containing descriptive language based on a subset of JSON notations, a tool to create and analyze the security plan on the cryptographic model and query and data validation based on the security plan. This can be very useful for the NoSQL database query processing. This method cannot be applied to the big data.

H. Yin, Z. Qin, L. Ou, and K. Li [13] design the system that enhance the privacy search scheme by allowing the user to create the random query trapdoor every time. The index was created for each data type by using the boolm filter and bilinear pairing operation. This helps the cloud to perform the search operation without using the useful information. This method proved to be the secure and extensive experiments demonstrate the correctness of the system. In the search efficiency, the pre-defined keyword is not produced, which is essential for the KNN.

H.I. Kim, H.J. Kim, and J.W. Chang [14] proposed the new secure query processing KNN algorithm in the cloud data. This method was designed to protect the both user data and query records. The index technique was used to increase the efficiency of the system without accessing the important data. The result showed that it reduced the query processing cost than the existing method with prevention of data. This research is focused on the query computational cost and it computational time is high.

L. Wang, Z. Yang, and X. Song [15] used the method called Secure and highly Available cloud database system in the Multi-Cloud (SHAMC) to secure the cloud. The method had secured the multi cloud data and homomorphic encryption was used to encrypt the data and queries was used on the cipher text. The entire dataset was stored in the multi cloud to avoid the interruption and also permanent failure and vendor lock-in. This method had the acceptable query load and it is superior to that of the other existing encryption dataset. It needs the data owner to do too many computations to perform encryption.

3. Proposed system

In this method, the data is encrypted by using the RSA method before stored in the cloud and decrypted by the private key. This section gives the overview of the proposed system, proposed RSA encrypted method, MD5, Load balancing and data integrity.

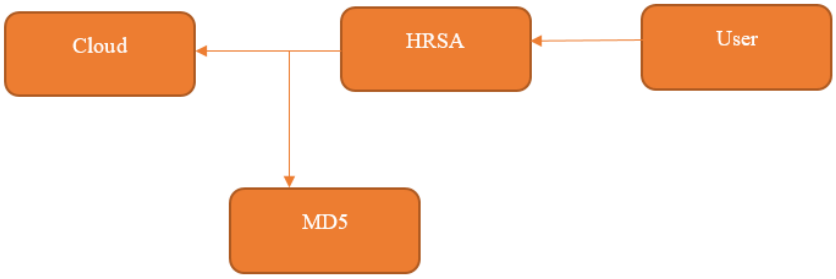

Figure.1 The Overview of proposed system

3.1 Overview

The overview of the proposed system is shown in Fig. 1, which shows the process of outsourcing the data to the cloud. The encryption is done with the help of RSA encryption and it is analysed by the MD5. The user uploads a data which is encrypted by RSA encryption and it has two keys: one is public key used to encrypt and another is private key, which is secrete key used to decrypt. Then MD5 analyse the encryption, decryption time and data integrity. The RSA is used to encrypt and decrypt the data with euglena function in the encryption.

3.2 RSA data encryption and decryption

Several methods were analysed to reduce the computational time of encryption and decryption process. HRSA security module is a public key encryption, which is used to encrypt and decrypt the data in the multi cloud environment. The RSA digital encryption efficiency is based on the large prime number used for key generation. The strong prime number is used to create the public and private key then it is used with Hamming weight and Euler's method to reduce the computational time. RSA method is based on the difficulty of the factorization of two large prime number and is measured as factoring problem. The equation of the HRSA security method is shown in equation.

1. Select two prime numbers p and q.
2. Calculate $n = p \times q$
3. Calculate $\emptyset(n) = (p-1) \times (q-1)$
4. Select the public exponent $e\varepsilon\{1,2,\ldots\ldots\emptyset(n)-1\}$ such that $\gcd(e,\emptyset(n)) = 1$.
5. Compute the private key d such that $d \times e \equiv 1 mod \emptyset$
6. Return $K_{+pub} = (n,e), K_{pr} = d$.

This steps involves in the encryption of the data by HRSA encryption method to the cloud. In this process $e,\emptyset(n)$ is the co-prime, p and q are the two large prime numbers and n is the private and public key module. RSA use two key in the encryption process they are: Public key and private key. The public key is visible to the user, it is used to encrypt the data and private key is not shown by the method, it is used to decrypt the data. The key generation of private key is shown in the equation.

$$d = e^{-1}(mod\emptyset(n)) \quad (1)$$

Where d represents the private key and e denotes the public key is shown in Eq. (1) and product of the private and public key is shown in Eq. (2).

$$d \times e = 1(mod\emptyset(n)) \quad (2)$$

And then a number h for hamming weight is consider in such a way that $\binom{n}{k} \geq -2H^2$ and $4H^2 \leq n \leq 16H^2$. From the derived key Hamming weight ration can be found as encryption key which can be derived as

$$pk = H = seq\left(\frac{int\,(d)}{int\,(e)}\right) \quad (3)$$

Encryption and Decryption of the cipher text is given in Eqs. (4) and (5) respectively, which is used in the process.

$$c = m^e(mod\ n) \quad (4)$$

$$m = c^e(mod\ n) \quad (5)$$

After the encryption of the data and it is uploaded to the cloud in the encrypted manner and it can access by the user. Most of the encryption system consists of the large module, which increases the power consumption and computational time. This method helps to reduce the length of the variable in the both encryption and decryption process. Thus increases the speed of the process and

optimize the system. This method is analysed by the MD5 method and compare it with the regular RSA system to analyse the efficiency of the proposed method.

3.3 MD5

MD5 uses the hash function with 128-bit hash value and it is used to analyse the data integrity of the encryption method. This is used to analyse the HRSA encryption method and compare it with the regular RSA method. The HRSA encryption system is analysed by the hash value by MD5. The procedure for the MD5 is given as follows:

3.3.1 Pseudocode for the MD5

i. Append 1 bit to message.
ii. Append 0 bit until message length in $bits \equiv 448$.
    a. Append original length in bits mod $2^{64}$to message.
iii. Initialize MD Buffer
    a. Var int a0= 0×67452301
    b. Var int b0= 0×efcd9b89
    c. Var int c0= 0×90badcfe
    d. Var int d0= 0×10325476
iv. Process message
    a. For each 512-bit chunk of padded message
    b. Break chunk into sixteen 32-bit words M[j], $0 \leq j \leq 15$.
        i. Var int A :=a0
        ii. Var int B :=b0
        iii. Var int C :=c0
        iv. Var int D :=d0
        v. For i from 0 to 63
            1. Var int F, g
            2. If $0 \leq i \leq 15$ then
                a. F : = (B and C) or ((not B) and D)
                b. g : = (5×i +1) mod 16
            3. else if $16 \leq i \leq 31$
                a. F : = (D and B) or ((not D) and C)
                b. g : = (5×i + 1) mod 16
            4. else if $32 \leq i \leq 47$
                a. F: = B xor C xor D
                b. g : = (3×i + 5) mod 16
            5. else if $48 \leq i \leq 65$
                a. F := C xor (B or (not D))
                b. g : = (7×i) mod 16
            6. F := F + A + K [i] + M[g]
            7. A := D
            8. D := C
            9. C := B
            10. B : = B + leftrotate (F, s[i])
        vi. End for
        vii. a0 : = a0 + A
        viii. b0 : = b0 +B
        ix. c0 : = c0 + C
        x. d0 : = d0 + D
    c. end for
    d. Var char digest [16]: = a0 append b0 append c0 append d0
    e. Leftrotate (x, c)
        i. Return (x << c) binary or (x >> (32-c));

3.4 Load Balancing

The large throughput is required for the big database and also requires the stability and scalability of the database. In this method, load balancing method is used to solve this problem in the cloud. The cloud database is divided into Read-only database and write-only database, if the cloud separation of database cost more. If the data owner consists of the large database, then it is necessary to divide the database. For example, E-bay deals with the trade all over the world and it have the read-write query ratio of 260:1. Therefore, it uses the read dataset higher than the write dataset, so it mostly need to execute the read-only database in order to balance the query pressure.

The database has been build and synchronize based on the idea of the replication. Multi cloud technique is used in this method, once user write queries then it synchronizes with the cloud. The database synchronize technique consists of two process statement-based synchronization and row-based synchronization. In statement-based

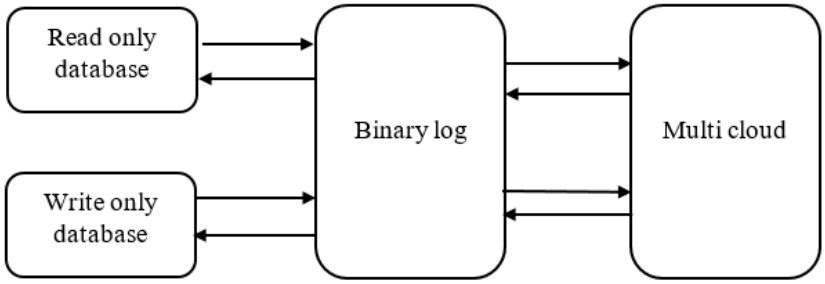

Figure.2 The cloud synchronize

synchronize technique, Data owner select the query that is converted into the binary log and the binary log is sent to the all the cloud to represent the query. In the row-based synchronize method, write-only cloud executes the query and changes on binary database into the binary log. The two types of synchronizing technique are shown in Fig. 2, which shows the both queries convert into binary log.

3.5 Data integrity protection

The cloud data should be taken backup for the prevention of the permanent failure in the cloud database. This is the viable solution to protect the data integrity and recover the cloud by using the backup data. Once the cloud suffers from the permanent failure, then data stored in the cloud database is paralyzed and it cannot be recovered. In this method, the data of the cloud is backup every time when data has been entered and helps to recover the data to the owner even after the permanent failure of the cloud. Every cloud has the separate copy of the encrypted data and it can be used if the cloud has suffered the permanent failure. Depending on the cloud synchronized to the backup cloud, each copy is valid and can be used for the data repair.

4. Experimental result

In this section, evaluation of the cost analysis, perceived availability and execution time of the proposed system is done. The proposed system has been designed with the client using the C# on the data owner side and MySQL is installed on the cloud side with protocols written by UDFs. The processor of two Intel Xeon CPU E5-4603 v2@2.20GHZ and 32GB RAM with running CentOS 6.5 in the 64-bit processor and MySQL 5.5. The proposed system used in the commercial manner and it evaluated in the practical environment. The four famous cloud server were used in this processed namely Ali's RDS, Microsoft's SQL Azure, Amazon's RDS, Tencent's CDB. This clouds are elastic and it can be adjusted for our environment and all clouds are running on MySQL 5.5.

The proposed system is processed and compared with the other existing system for the data base workload evaluated for 30 minutes. The parameters like Read queries, write queries, transaction, deadlock, Transaction per second (TPS) and success rate of the given system. It is compared with other existing systems such as SHAMC[2], SHAMC[3], SHAMC[4], CryptDB, MONOMI, SDB, Amazon RDS [15]. This clearly shows that the HRSA outperforms the all other system in the success rate. This method can be used to encrypt the data before upload it to the cloud. The Hamming weight helps to generate the key in the simple manner and security is improved. This doesn't take much resources in the system and uses the most of the resources for the executing the other purpose. So that the success rate is increased and TPS is also increased. The performance of the system is increased, so the deadlock is decreased.

Development costs are needed to be reduced in order to make commercially success. To analyse the cost of this method, we evaluated the typical scenario of the enterprise 1TB hard disk, 24 GB of RAM, 80 Mbps bandwidth and 8-core processor. The system with the two cloud performance well in the range of cost for the year without any additional investment. If the availability of the cloud is high, then the multi cloud can be used.

The proposed system increases the availability of the cloud and it shown with the other existing system. The cloud availability of the system is measured through the database workload test Sysbench. The TPS and success rate of the various system for the cloud encryption is shown in Fig. 3. The success rate measures the successful transmission of the method and not every transmission is succeeding in the process. The deadlock may occur in the request of queries to the cloud and this is common in the large database, which cannot be fully avoided. This graph clearly shows that proposed system have the higher transmission than the existing system. The second

highest transmission is Amazon RDS. The SHAMC4 [15] have the third highest transmission rate and it also have the higher success rate. The proposed HRSA outperforms the all other existing system in both TPS and success rate, due its design and multi cloud architecture. The performance of the system can be increases by increasing the number of cloud in the method.

Table 1. Evaluation of database workload (30 minutes)

	SHAMC² [15]	**SHAMC³ [15]**	**SHAMC⁴ [15]**	**CryptDB [15]**	**MONOMI [15]**	**SDB [15]**	**Amazon RDS [15]**	**HRSA**
Read Queries (10^7)	1.1955	1.3518	1.4212	0.7826	0.9856	0.7521	1.3996	1.4237
Write Queries (10^7)	0.2152	0.2549	0.2864	0.1056	0.1685	0.1041	0.4665	0.5212
Transaction (10^5)	7.6512	7.9691	8.1251	6.0519	6.5849	6.0551	9.9882	10.1254
Deadlock	1265	1143	1095	1521	1582	1438	846	698
TPS	7468	7823	8512	5125	5816	5049	10659	11427
Success Rate (%)	99.84	99.86	99.87	99.75	99.76	99.76	99.92	99.94

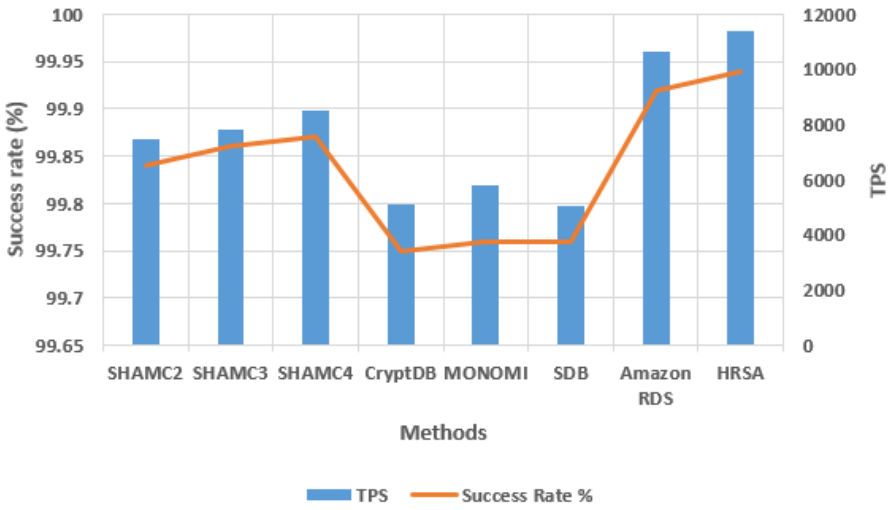

Figure.3 TPS and Success Rate

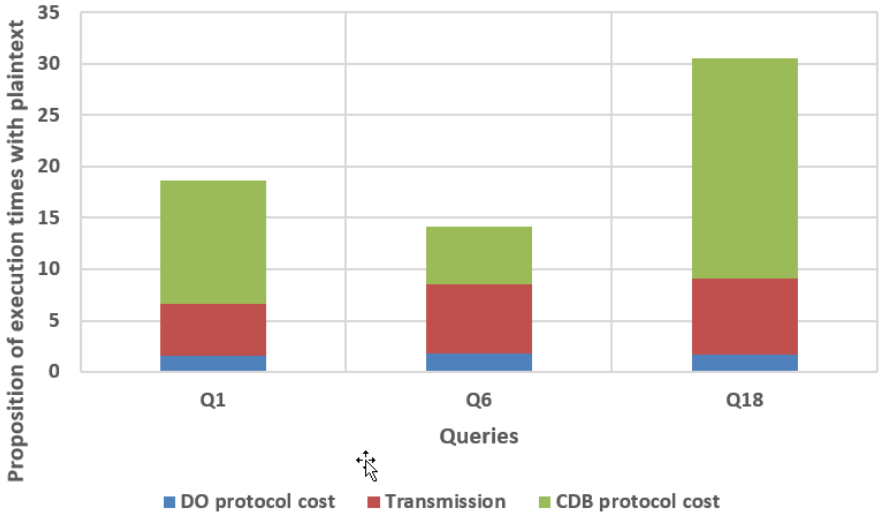

Figure.4 Graphical representation of the execution times with queries

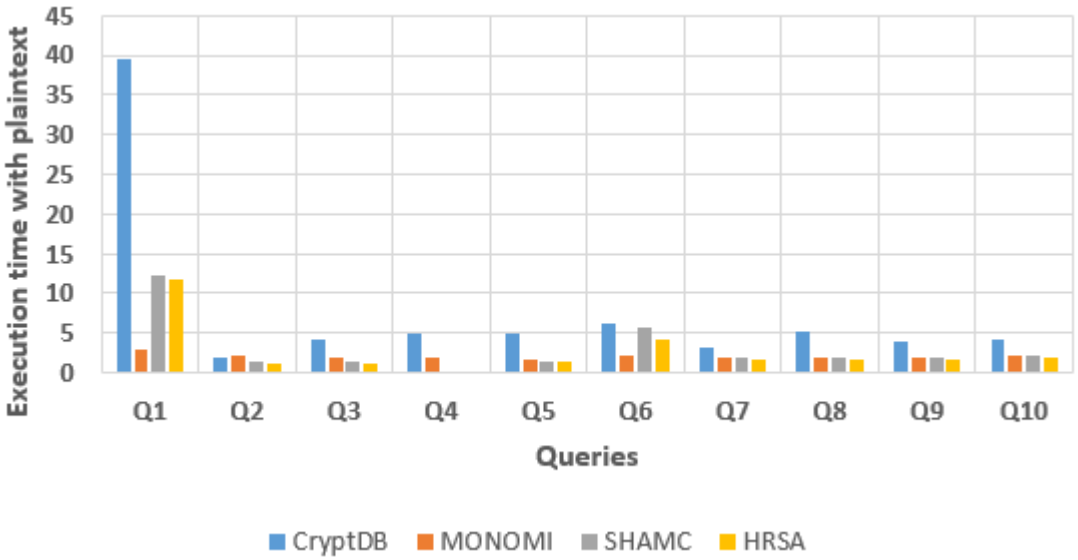

Figure.5 Represent the execution time with the plaintext

The execution time of the three queries has been calculated with different cost and compared with each other. The Cloud Database (CDB) protocol cost have the higher transmission cost compare to the other two cost. TPC-H benchmark is used for the testing large database and queries has been executed with the higher degree of complexity. The query execution is divided into three parts namely Data Owner (DO), Transmission and CDB.

The proposed system is processed for the data in the cloud to encrypt and decrypt with less computation time. The TPC-H benchmark plain text database is processed with the proposed system and execution time is calculate for each queries. The graphical representation of the HRSA process of the different queries and their respective execution time.

5. Conclusion

Cloud computing have the lot of facility for the user and user have many advantageous in the cloud computing. In present day usage of cloud computing increases highly, so more secure and effective cloud is required for the user. Lot of encryption method is presented for the data in the cloud and most of the method take the high computational time. In this research, HRSA method is used in the cloud for the encryption of data with less computational time. The encryption and decryption time of the HRSA system is reduced due to low complex value is used in the method. This method creates the two key namely public and private key for the system. The public key is visible to the user and it is used to encrypt the system, whereas private key is hidden and used in decryption process. The encryption and decryption is done with the different key and MD5 hash value is used to analyse the system. The encryption and decryption key is generated with the limited computational resources and this increases the performance of the system in terms of success rate, TPS and reduces the deadlock. A lot of resources are available for processing, which reduces the deadlock of the system. The performance of the proposed system was compared to the other existing systems like SHAMC[2], SHAMC[3], SHAMC[4], CryptDB, MONOMI, SDB and Amazon RDS. The success rate of the system is 99.94% and deadlock reduced to 698 times. The experimental result clearly showed that the proposed method reduced the computational time compared to the other existing methods.

The future work will focus on the external security issue of this method. This method only focus on the internal methods like success rate, data loss etc.

References

[1] S.F. Bailey, M.K. Scheible, C. Williams, D.S. Silva, M. Hoggan, C. Eichman, and S.A. Faith, “Secure and robust cloud computing for high-throughput forensic microsatellite sequence analysis and databasing”, *Forensic Science International: Genetics*, Vol.31, pp.40-47, 2017.

[2] S. Lallali, N. Anciaux, I.S. Popa, and P. Pucheral, “Supporting Secure Keyword Search in the Personal Cloud”, *Information Systems*, Vol.72, pp.1-26, 2017.

[3] Q. Huang, Y. Yang, and M. Shen, “Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing”,

Future Generation Computer Systems, Vol.72, pp.239-249, 2017.

[4] H. Mezni and M. Sellami, "Multi-cloud service composition using Formal Concept Analysis", *Journal of Systems and Software*, Vol.134, pp.138-152, 2017.

[5] T. Xiang, X. Li, F. Chen, S. Guo, and Y. Yang, "Processing secure, verifiable and efficient SQL over outsourced database", *Information Sciences*, Vol.348, pp.163-178, 2016.

[6] Y. Peng, J. Cui, H. Li, and J. Ma, "A reusable and single-interactive model for secure approximate k-nearest neighbor query in cloud", *Information Sciences*, Vol.387, pp.146-164, 2017.

[7] K. Kritikos, T. Kirkham, B. Kryza, and P. Massonet, "Towards a security-enhanced PaaS platform for multi-cloud applications", *Future Generation Computer Systems*, Vol. 67, pp.206-226, 2017.

[8] L. Wu, B. Chen, K.K.R. Choo, and D. He, "Efficient and secure searchable encryption protocol for cloud-based Internet of Things", *Journal of Parallel and Distributed Computing*, vol. 111, pp.152-161, 2018.

[9] P. Li, J. Li, Z. Huang, T. Li, C.Z. Gao, S.M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing", *Future Generation Computer Systems*, Vol.74, pp.76-85, 2017.

[10] R. Xu, K. Morozov, Y. Yang, J. Zhou, and T. Takagi, "Efficient outsourcing of secure k-nearest neighbour query over encrypted database", *Computers & Security*, Vol.69, pp.65-83, 2016.

[11] C. Huang, R. Lu, and K.K.R. Choo, "Secure and flexible cloud-assisted association rule mining over horizontally partitioned databases", *Journal of Computer and System Sciences*, Vol.89, pp.511-63, 2016.

[12] M. Ahmadian, F. Plochan, Z. Roessler, and D.C. Marinescu, "SecureNoSQL: An approach for secure search of encrypted nosql databases in the public cloud", *International Journal of Information Management*, Vol.37, No.2, pp.63-74, 2017.

[13] H. Yin, Z. Qin, L. Ou, and K. Li, "A query privacy-enhanced and secure search scheme over encrypted data in cloud computing", *Journal of Computer and System Sciences*, Vol.90, pp.14-27, 2017.

[14] H.I. Kim, H.J. Kim, and J.W. Chang, "A secure kNN query processing algorithm using homomorphic encryption on outsourced database", *Data & Knowledge Engineering*, In press, 2017.

[15] L. Wang, Z. Yang, and X. Song, "SHAMC: A Secure and highly available database system in multi-cloud environment", *Future Generation Computer Systems*, In Press, 2017.