International Journal of Scientific Engineering and Technology (ISSN: 2277-1581)
Volume No.2, Issue No.3, pp: 122-128 1 April 2013

# Survey of Confidentiality and Integrity in Outsourced Databases

**Ajeet Ram Pathak[1], B. Padmavathi[2]**
[1,2]Department of Computer Engineering,
University of Pune,
M.S., India
Email: [1]ajeet.pathak44@gmail.com, [2]bpadma_cse@yahoo.com

***Abstract-- Due to advancement in the Cloud Computing technology, Database as a Service (DBaaS) - a novel data management and administration paradigm has gained the worldwide popularity. The data owner outsources their confidential data to the third party service provider's site.***

***The service provider manages and administers the database and avails the readymade services to the data owner and their clients. Though the database services are provided to the clients, the security of outsourced databases is the most prominent issue to be handled since the database is under the control of distrustful service provider. Various security services like confidentiality, integrity, availability, authenticity, accounting, etc are needed to be incorporated into outsourced databases. These security services are fulfilled by implementing the encryption based approaches, data distribution based approaches, indexing techniques, fake tuple insertion approaches, signature schemes, authenticated data structure approaches. The thorough analysis of all these security techniques along with their significance is given in this survey paper.***

***Keywords—Confidentiality, Database as a Service, Integrity, Outsourced Databases***

## 1. Introduction

The enormous amount of generation of data in business firms, scientific organizations and education institutes leads to the need of data management and administration mechanism. This need is fulfilled by novel data management approach namely – Database as a Service. In this paradigm, the data owner outsources the private data to the third party service provider. The service provider is delegated the responsibility of managing and administering the database. At the service provider's site, all the necessary software and hardware tools are deployed. The service provider allows the data owner and respective clients to insert, update and delete the data. All the transaction management, data backup and recovery mechanism, fault tolerance, disaster management is done by the service provider. Due to this, the company can invest their valuable assets for core business logic only since the database management is done by third party service provider.

Preserving the security of the outsourced databases is a great challenge in the current scenario. According to the data breach investigation done by Trustwave in 2012, 76% of security deficiencies were caused by the third party service provider [1]. Therefore, it is very crucial for the companies to be aware about security enforcement in their outsourced databases to keep the data confidential and thereby complying with the government rules and regulations.

*Confidentiality*, *integrity* in context of *completeness* and *correctness*, *authenticity*, *accountability*, etc are considered as the pillars of security services. Therefore, implementing them in an efficient manner is very important from the security point of view. Various techniques are used for realizing the security in database outsourcing. These techniques include encryption, authenticated data structures, indexing, signature schemes, etc. In this paper, we have given the complete analysis of security techniques along with their pros and cons.

The paper is arranged in a following way: The overview of database outsourcing including the architecture and benefits of outsourcing is given in section 2. The section 3 deals with general security requirements for outsourced databases. The security techniques are analyzed in section 4. The paper is concluded in section 5 along with the future scope.

## 2. Preliminaries

The concept of DBaaS, architecture models of database outsourcing and its usefulness is discussed in this section.

### A. Concept of Database as a Service

According to the Wordpress [2], DBaaS is defined as,

*"A managed service, offered on a pay-per-usage basis, that provides on-demand access to a database for the storage of application data."*

The DBaaS ameliorates the need to purchase and install the data management hardware and software at the data owner's site. The data owner and clients use the readymade database service availed to them by service provider.

The Organizations pay for the database service they are getting from the service provider. For the companies with less amount of resources limited hardware and time-bound projects, DBaaS best suits the scenario. Due to its inherent scalable property, DBaaS can scale up well in case of increasing user demands and also scale down when the demand subsides. The deployment of infrastructure for industries gets easier with the help of DBaaS. It offers flexible and on-demand services, optimizes performance tuning of the system, lowers the operating cost and complexity, accelerates the provisioning i.e. allows to clone the old database with a new schema, shortens the sales cycle, provides failover environment for project execution, enables the centralized administration and management of all kinds of databases.

### B. Architecture Models of Outsourced Databases

Generally Speaking, 3 entities are involved in database outsourcing viz. data owner, clients and service provider. The data owner is responsible for uploading the data at the service provider's site. The data owner has the sole authority to permit or deny the clients for accessing the database. The clients are also called as queriers who access the database according to the privilege level acquired by them. The database service can be availed through portable devices like PCs, tablets and smartphones. The service provider performs all the data maintenance tasks. For efficient data communication, the transmission link between the service provider and the data owner as well the link between service provider and the clients should be of high bandwidth.

There are three kinds of architectural models of outsourced databases. The figure 1 shows the *uniform client model*. In this model, the data owner and the client are the same. The data owner performs all the operations on database (Insert, update, delete). This is the simplest database outsourcing model. The *Multiple client outsourced database model* is depicted in figure 2. This model comprise of single data owner and multiple clients (queriers). The *Access control* mechanism is implemented in this model for the clients according to their privilege-level. In *multiple data owner database outsourcing model* shown in figure 3, each data owner uploads the data at the service provider's site. In this individual *access control* policies for each group of data owner and clients are needed to be implemented. The Attribute based access control, accounting and authorization are some mechanisms which can be incorporated in *multiple data owner model*.

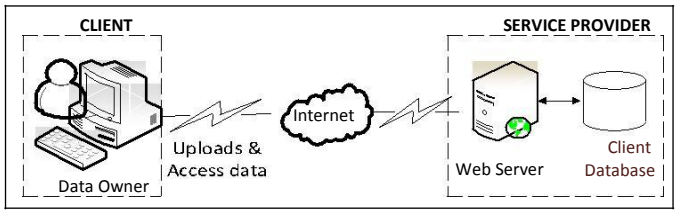


Fig. 1. Unified Client Model

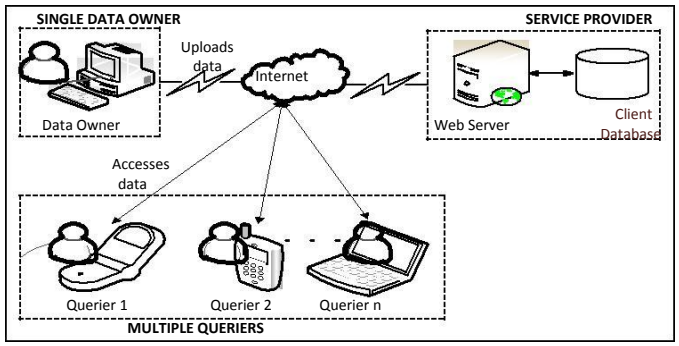


Fig. 2. Multiple Client Model

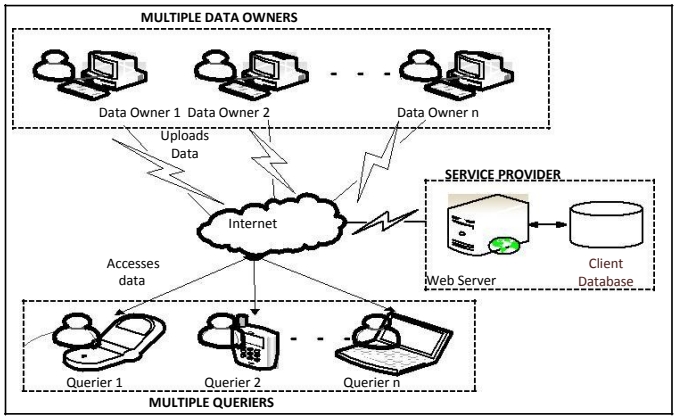


Fig. 3. Multiple Data Owner Model

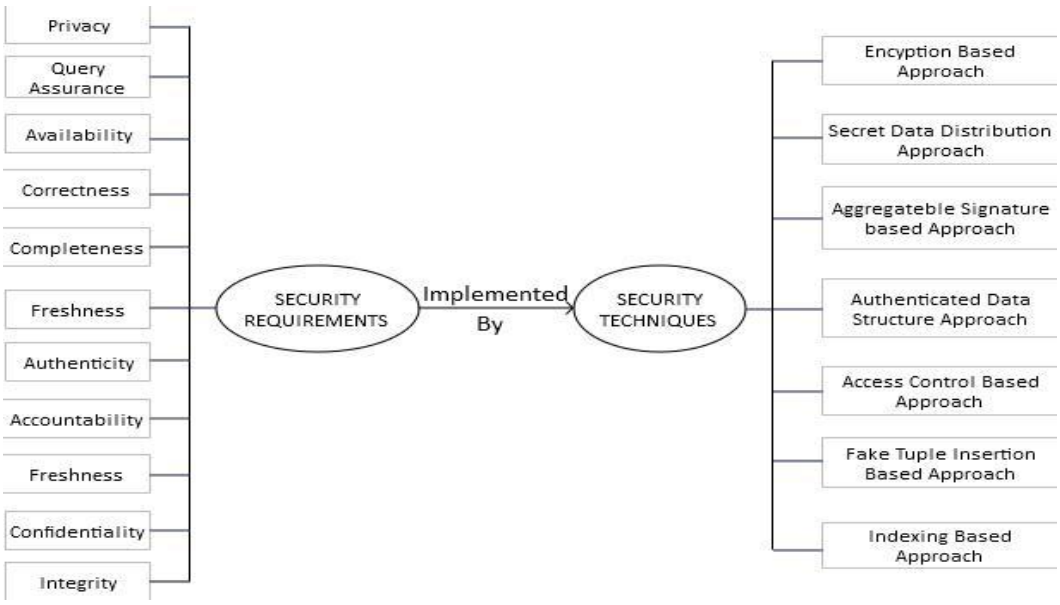


Fig. 4. General Security Requirements and Techniques for achieving Security in Database Outsourcing

## 3. General Security Requirements

The figure 4 depicts the general security requirements and the techniques for implementing them in database outsourcing.

The *confidentiality* is one of the important aspects in security. Making the data unintelligible when it is in the transit state or stored in data center is referred to as preserving the *data confidentiality*. *Privacy* is also considered while maintaining the *confidentiality*. Generally, the *privacy* comprises of *user privacy* and *access privacy*. For hiding the identity of the user, the *user privacy* is considered. *Access privacy* conceals the database access pattern for a particular user.

*Integrity* assures that the data being stored in the database or being transmitted in the network is tamperproof or unaltered.

*Integrity* can be considered as the combination of two dimensions as *completeness* and *correctness*. The *completeness* guarantees that the query results are retrieved by executing the query over all the database records which contain the predicate (tuple) expressed in the query.

*Correctness* promises that the results gained by executing the query against the database are unaltered, correct and are produced by the genuine database servers or genuine processes accessing the database. *Query assurance* lets the client believe that query is executed over the genuine database server only. *Availability* is important aspect in the security triad of *CIA (Confidentiality, Integrity, Availability)*.

*Availability* is defined as the degree to which the database system is up and working in an operable state. It is very much crucial for the service provider to make the database service available all the time. The *availability* is calculated as:

=

+

Where, refers to the time during which the system is available for use while is the time required by the system to recover (Time during which the system is not in operable state).

For relational databases being outsourced, *availability* is inherently supported. But in case of NoSQL databases, *CAP (Consistency, Availability and Partition Tolerance) theorem* [3] is applied. The *CAP theorem* is depicted in figure 5. The consistency assures that the client has always the same view of data. *Availability* means the client gets an access to the database seamlessly and without interruption. Partition Tolerance states that system works well even if is physically partitioned. For NoQL databases, only two parameters from *CAP theorem* are needed to be picked up. The pair {*Consistency, Availability*} *(CA)* is supported by relational databases while the pair {*Consistency, Partition Tolerance}* *(CP)* is supported by MongoDB, HBase, Redis, etc databases.

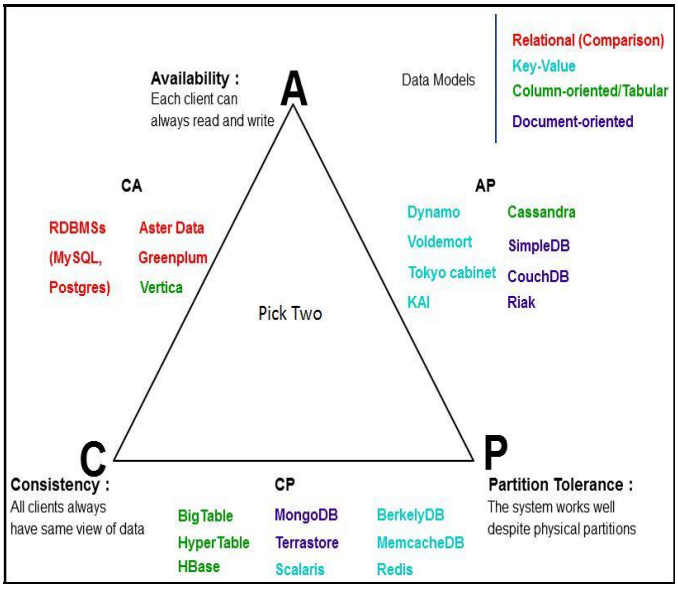


Fig. 5. CAP Theorem

Cassandra, Dynamo, CouchDB, etc databases supports the {*Availability and Partition Tolerance*} *(AP)* features from the *CAP* theorem. In case of the databases with {*CP*} feature, the *availability* of databases might get hampered.In context of information security for outsourced databases, *authenticity* refers to the trustworthiness and genuineness of databases, communication via transmission links, transactions, clients, data owners and the service provider. All the entities must be validated for ensuring the authenticity. Digital signature provides the better way to achieve the authenticity.

*Freshness* is the new aspect considered in database outsourcing. *Freshness* of database is assured only when the query is executed on the most recent edition (version) of the database uploaded by the data owner. Maintaining the *Freshness* has a great significance when the database is continuously or periodically updated and upgraded by the data owner. By sending the timestamp to the clients showing the validity of database is a good approach for ensuring the *Freshness* of the database.

Every entity is delegated the privilege level to access the data. The tasks performed by each entity are accountable for that entity only. This is called as *accountability*. *Access control* is referred to as allowing only the authorized users to access the protected data they are permitted to.

*Access control* can be realized by following the three steps viz. *Identification*, *Authentication* and *Authorization*. Identification is the act of finding which entity is querying the system. Once the identification is completed, authentication comes into picture. It refers to verify the claim of an entity to be genuine. For implementing the robust security, the multi-factor authentication mechanism can be implemented. The multi-factor authentication can be the combination of username, passwords, biometric authentication and the unique assets like swipe cards. Once the specific entity is identified and authenticated, which data is permitted to access and which kinds of operations on data (Read, Write, Execute, Update) are allowed to be performed is found out. This is called as authorization.

There are mainly three kinds of *access control* models. These models are discretionary access control model (DAC), mandatory access control model (MAC), and the role-based access control model (RBAC). The DAC model allows the prescribed users to access the permitted objects (database, protected entities) by identifying the user or the group to which he belongs. Access control matrix, Access control list, Access control capabilities lists are some mechanisms for implementing the DAC model.

The MAC model is used for controlling the information flow of the system. The access control policy in MAC is harder than that in DAC. The access authority is used to define the role of user in RBAC model. The user is given a permission to access the data according to his role and privilege level.

## 4. Detailed Discussion of Security Techniques in Database Outsourcing

The figure 4 shows the techniques for achieving the security in outsourced databases. The detailed analysis of these techniques is done in this section.

For preserving the *privacy* and *confidentiality* of outsourced databases, encrypting the databases is found to be an inherent option [4]-[11]. Both the software and hardware level encryption is put forth in [4] for IBM DB2 database. The software level encryption is realized using the RSA (Rivest, Shamir and Adleman) algorithm and Blowfish algorithm. The blowfish algorithm outperforms the RSA algorithm. The specialized encryption hardware is used to encrypt the

database at a row level. The query execution time is very less for the hardware level encryption compared to software level.

The *privacy* is achieved using the *confidentiality* constraints in [5]. In this, the secret data is split up into fragments so as to reduce the secret association (affinity) between the attributes of data. These fragments are encrypted and stored on different servers. The encryption key is only known to the trusted entities like data owner and the trusted clients. To maintain the fragmentation level, a heuristic approach is implemented. This approach does not support the dynamic updates as it becomes very difficult to change the data stored in fragments efficiently.

Encrypting the data for achieving security and decrypting it for retrieving the results causes extra computation overhead on the system. This lowers the system performance. Rather than decrypting the data, there is one option of executing the query over the encrypted database which is put forth in [6], [7], [8].

In case of distrustful server, to assure the *confidentiality* of data at the service provider's site, a mechanism is put forth in [6] in which the SQL query is executed on the encrypted data. In this, the query is divided into two parts namely, client query and the server query. The service provider executes this server query and gives encrypted rows as an output to the client. At the client side, client executes the remaining query and by decrypting it, obtains the results; thus ensuring the *privacy*. The drawback of this approach is that it increases the computation cost at the client side. As the encrypted rows are transmitted via network link, computation overhead is high.

Querying the encrypted data is inefficient and inaccurate. For achieving the efficiency in executing the query encrypted data, Order Preserving Encryption (OPE) is proposed in [9], [10], [11].

To maintain the order between the plain text values and cipher text values, OPE is useful. OPE works as follows [9], [10]. Let us say, $F$ is an OPE function. $P_1$, $P_2$ and $C_1$, $C_2$ are plain text and cipher text values respectively such that $C_1 = F(P_1)$ and $C_2 = F(P_2)$. If $P_1 < P_2$ then $C_1 < C_2$. These approaches do not withstand plain-text chosen attack. To make this approach more efficient, noise based OPE approach is proposed in [11]. In this, secret value $x$ is mapped to the linear equation containing the *noise*. The *noise* value is incorporated into an equation such that order of input value is retained. This approach is brute-force attack tolerant. But, it supports range query only. The main advantage of encryption is that it makes the data unintelligible. It is useful for small databases. In case of large databases, encryption and decryption causes extra overhead on the system leading to degradation in system performance and efficiency. Encryption based techniques suffer from key management overheads.

The secret data distribution approach alleviates the need of encryption. Shamir's (k, n) threshold Secret Sharing scheme is the backbone of secret data distribution approach [12]. According to secret scheme, the Data $D$ to be protected is divided into n parts as $D_1, D_2, \ldots D_n$ and distributed among n entities. This data is called as secret share. Only $k$ or more $D_i$ data parts can reconstruct the secret. From $k-1$ or lesser $D_i$, the secret can't be revealed.

The *integrity*, *completeness* and *Freshness* are collectively achieved in [13] based on secret sharing scheme. In this, the secret data is represented using the polynomial and this polynomial function is distributed among $n$ number of data server. In this paper, an indexing based approach is used for the Index Server. The index server is used for referencing the data stored in the data servers. The $B^+$ tree is created over the searchable attributes. The leaf of the $B^+$ tree corresponds to the encrypted bucket which contains the Tuple ID for the record. The aggregate-able signature scheme namely – condensed RSA is used by the data owner for *Freshness* verification. For calculating the secret values when a client fires the query, the Lagrange's Interpolation formula is used only if $k$ data servers respond to the client. Thus, provides the collusion resistance in the system. This approach supports all the queries including range, equality, projection and aggregation query. But it only supports to give security to the numeric data. To achieve the authentication and integrity, the condensed RSA is used as an aggregate-able signature scheme in [14]. This approach can withstand the adaptive chosen message attack.

Merkle B tree – dynamic authenticated index structure is adopted in [15] to provide the *Correctness*, completeness and *Freshness* guarantees. It is the combination of Merkle hash

tree and the $B^+$ tree. In this approach, the client needs to calculate the hash of the root by traversing the tree completely. As the client traverses the whole tree, completeness verification is achieved. The signed hash root value is checked with the value obtained by the owner's public key. If both the values are same then the *Correctness* is verified. For

*Freshness*, in case of each database update, signature is re-issued showing the validity of data *Freshness*. This approach supports the static as well as the dynamic updates. The drawback of this approach is that multi-dimensional structures are not supported.

For achieving the fine-grained *access control* in the multiple data owner outsourced database model, a Public Key Infrastructure management scheme is adopted in [16]. This approach also addresses the issue of scalability in case of continuously changing users. The multi-agent system model is developed for multi-policy enforcement. The verification scheme of PKI is used for maintaining the integrity data access and the communication done for resource sharing. The accountability is also supported in this approach by tracing the user request for data using the timestamp.

To ameliorate the need of public key cryptography and thus reducing the storage requirement, dynamic *access control* method is proposed in [17]. The polynomial function and double encryption are the basis of this approach. The advantage of this scheme is that it transparent to the users. The users are not involved in the procedure of *access control* model when they are added into system or exited from system. Addition or deletion of users does not cause re-encryption of the resource and re-updating the decryption keys. The client has less computation overhead as the major computation load is shifted to the data owner side. For each user, only the two keys are needed to be maintained so storage overhead is also low. This approach does not require any interaction among the involved entities.

An encryption scheme using the multiple proxies supporting the multiple users is used to provide the *data privacy* in given in [18]. Each user is allotted a distich storage space and the query key pair for accessing the database which hides the query fired by the user and query result obtained.

For multi-user and multi-application based environment, the Role Based Access Control (RBAC) method based on Public key Infrastructure management and X.509 Privilege Management Infrastructure is put forth in [19]. The trio authentication, authorization and accountability are supported in this approach. The combination of dynamic RBAC and attribute based policy driven *access control* is contributed in [19]. This system lacks the robustness in terms of agent recovery. The approach does not support the scalability for acquiring large number of clients.

The multi-user *access control* for the encrypted cloud based databases is given in [20]. The key policy based attribute based encryption is adopted to allow only the authorized users to access the data. This policy is associated with the user's decryption key and the attributes in the policy are associated with the cipher text. The user is given reading access to the database only if the KP-ABE key allows it.

## 5. Conclusion and Future Scope

The Database as a Service is a recent database management solution which is becoming popular day by day due to its usefulness. Implementing the security in the outsourced databases has been the prominent issue till date. In this paper, we have explained the concept of DBaaS and its benefits. The architectural models of the outsourced databases are also elaborated. The thorough analysis of general security requirements for the outsourced databases is done in this paper. The survey of the security techniques put forth for achieving the security techniques in outsourced databases is portrayed. The detailed discussion of achieving the *confidentiality*, integrity, completeness, *Correctness*, *access control* and accountability in single and multi-user environment is given

The generalized security framework can be developed such that it supports all types of databases and all the types of queries. The alternative techniques for the encryption can be found out to enhance system performance. The investigation on securing the network link for protecting the data sent on the link can be done and implemented to defeat the eavesdropping. The future enhancement can be focused on optimizing the query processing and the communication cost.

## References

[1] http://www.networkworld.com/news/2012/020712-data-breach-255782.html

[2] http://dbaas.wordpress.com/2008/05/14/what-exactly-is-database-as-a-service/

[3] http://en.wikipedia.org/wiki/CAP_theorem

[4] H. Hacigumus, B. Iyer and S. Mehrotra, "Providing database as a service," in Proc. of IEEE 18th ICDE, 2002, pp. 29-38.

[5] V. Ciriani, S. D. Vimercati, S. Foresti, and S. Jajodia, "Combining Fragmentation and Encryption to Protect Privacy in Data Storage," TISSEC, vol. 13, pp. 1094-9224, 2010.

[6] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over Encrypted Data in the Database-Service-Provider Model," SIGMOD'02, USA, 2002, pp. 216-227.

[7] Z. F. Wang, J. Dai, W. Wang, and B. L. Shi, "Fast Query over Encrypted Character Data in Database," Computational and Inf. Science, vol. 4, pp. 1027-1033, 2005.

[8] Zheng-Fei Wang, Ai-Guo Tang, "Implementation of Encrypted Data for Outsourced Database", In Proc. of Second International Conference on Computational Intelligence and Natural Computing (CINC), IEEE, 2010, pp. 150-153.

[9] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order pre-serving encryption for numeric data. In Proc. of the 2004 ACM SIGMOD international conference on Management of data, SIGMOD '04, pages 563–574, 2004.

[10] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In Proceedings of the 28th Annual International Conference on Advances in Cryptology, EUROCRYPT '09, pages 224–241, 2009.

[11] Dongxi Liu, Shenlu Wang, "Programmable Order-Preserving Secure Index for Encrypted Database Query" In Proc. of 2012 IEEE Fifth International Conference on Cloud Computing, pp. 502-509, 2012.

[12] A. Shamir. "How to share a secret" In Communications of the ACM, 1979, pp. 612–613.

[13] M. A. Hadavi, M. Noferesti, R. Jalili, E. Damiani "Database as a Service: Towards a Unified Solution for Security Requirements" In Proc. of IEEE 36th International Conference on Computer Software and Applications Workshops, pp. 415-420, 2012.

[14] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases", In Proc. of ACM Transactions on Storage, vol. 2, 2006, pp. 107-138.

[15] Li Feifei, Marios H, George K, "Dynamic Authenticated Index Structures for Outsourced Database", In Proc. of ACM SIGMOD'06. Chicago, Illinois, 2006, pp. 121-132.

[16] Somchart Fugkeaw, "Achieving Privacy and Security in Multi-Owner Data Outsourcing", In Proc. of IEEE Transactions 2012, pp. 239-244.

[17] Xiaoming Wang , Yanhui Zhang, "A Dynamic Access Control Scheme for Outsourced Database" In Proc. of International Conference on Network Computing and Information Security, 2011, pp. 1-7.

[18] Nguyen Thanh Hung, Do Hoang Giang, Ng Wee Keong, Huafei Zhu, "Cloud-Enabled Data Sharing Model" In Proc. of IEEE International Conference on Intelligence and Security Informatics (ISI), 2012, pp. 1-6.

[19] Fugkeaw, Manpanpanich, Juntapremjitt, "Exploiting X.509 Certificate and Multi-Agent System Architecture for Role-based Access Control and Authentication Management" In Proc. of IEEE International Conference on Computer and Information Technology, CIT 2007, pp. 733 - 738.

[20] Mihaela Ion, Giovanni Russello, Bruno Crispo, "Enforcing multi-user access policies to encrypted cloud databases" In Proc. of IEEE International Symposium on Policies for Distributed Systems and Networks, 2011, pp. 175-177.