IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 5 (Jul. - Aug. 2013), PP 35-42
www.iosrjournals.org

Secure Access to Outsourced Databases

Sanjeev Kumar Chauhan[1], Aravendra Kumar Sharma[2]
[1](Department of Computer Science and Engineering MNNIT Allahabad, India)
[2](School of Computer Science and Engineering, Galgotias University Greater Noida, India)

Abstract : *Data security is one of the fundamental security requirement for outsourced databases in a cloud computing environment. Existing solutions usually suffer from problems such as information leakage, key management, revocation handling and user authentication. In this paper to overcome these security challenges, the proposed scheme uses a modified row- based encryption technique where each row of the database is encrypted using a separate key. The proposed scheme also uses client side memory, and selective encryption to increase performance of the system. The proposed approach also uses a modified Diffie-Hellman key exchange algorithm for establishing secure communication between cloud service provider and the user, inhibiting malicious outsiders.*

Keywords - *authentication, client side memory, key management, Outsourced databases, row-based encryption, selective encryption.*

I. INTRODUCTION

Cloud computing provides several services to it's clients at minimal cost. It provides services categorized as follows: Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS). Database as a service (DaaS) is basically a Platform as a service which is also known as Database outsourcing. Recently Database outsourcing received considerable attention in cloud computing. Due to ease in maintenance and security characteristics it has become a cost effective and viable alternative to traditional in-house database management. In "Database-as-a-Service" (DaaS) model, a Database Owner delivers its database to the Cloud Service Provider who in turn provides the clients seamless access to database [2]. However, database outsourcing poses several security problems related to data confidentiality, integrity, privacy and authentication [12] [13] [21]. Sometimes data are so confidential that the Database Owner prefers to avoid disclosing them even to the Cloud Service Provider. In the past, database encryption has been proposed as a solution. But encryption poses several problems related to access policies, key management [14] and revocation handling.

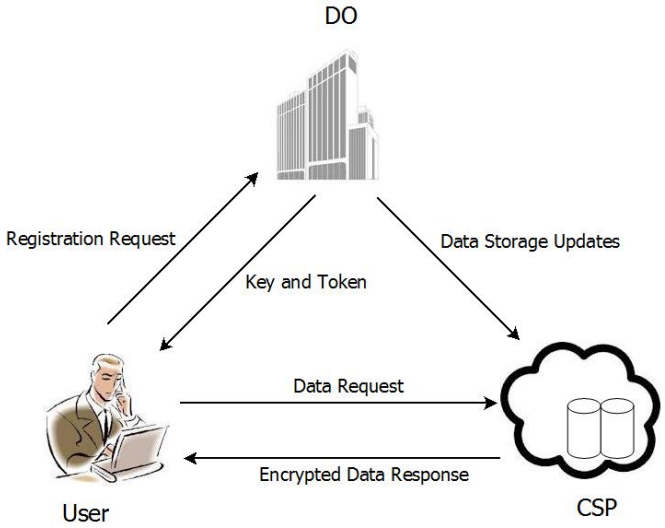

Fig.1. Traditional Database Outsourcing Model.

The steps for querying outsourced databases in a cloud computing environment are described in Fig 1. Initially, the user registers itself with the Database Owner, who in turn provides access privileges to the user. These changes are conveyed to the database residing at the Cloud Service Provider. Once changes are updated to the database, user can communicate directly to the Cloud Service Provider for data access.

Databases can be encrypted at various levels of granularity such as, whole database, table, column or row [3] [5] [15]. Whenever whole database or a table is encrypted using a single key, it makes schema definition complex and security directly related to the degree of confidentiality with which the key is kept. Whenever column-based encryption is used it is not possible to assign access privileges to users. Instead, row-based encryption ensures better access control. However, this technique has significant disadvantages like constraints and triggers become almost unusable. To solve these constraints proposed scheme uses a modified row-based encryption mechanism which is highly secure, effective and has ease in accessing database.

The rest of the paper is structured as follows. Section II reviews previous researches related to this paper. Section III discusses the proposed model and assumptions. Section IV describes the proposed scheme. Section V analyses effectiveness and performance of the proposed model. Section VI presents simulation details and results. Finally, section VII concludes the paper.

II. RELATED WORK

Many researchers are working for securing outsourced databases in cloud computing. Hakan Hacigumus [1] was the first to propose "Database as a Service" also known as Database Outsourcing. Their after, lot of schemes are proposed to secure outsourced databases [20] [22]. Most of the proposed schemes preferred encryption as a solution for implementing security on outsourced databases. Database is encrypted at various levels of granularity such as, whole database, table, column or row [3] using secret keys. However, use of secret key has risen key management problem [2] [18] i.e. managing key distribution [16] and handling revocation.

Bennani [2] proposed a solution to reduce key management complexity by sharing secret key between user and set of servers on the cloud and for key distribution a role based approach was introduced. In Bennani protocol, database is replicated n times for each role. Where each copy of the database is encrypted using a virtual key α (as a set of shadows S_1, S_2 .....S_n). Each shadow is placed with the different copies. User encrypts his query with the key β and send $\beta(q)$ to CSP. CSP in turn encrypts $\beta(q)$ with each shadow. Later on, $\alpha(\beta(q))$ is decrypted using β by the user and then requested query $\alpha(q)$ is executed on database by the CSP, and the response $\alpha(r)$ is sent back to the user. Then $\alpha(r)$ is again encrypted by the user using key β and $\beta(\alpha(r))$ is delivered to the CSP, who in turn decrypts $\beta(\alpha(r))$ with each of the shadows and the resulted response $\beta(r)$ is sent back to the user, who in turn decrypts it and has access to the results. Although the scheme is good, but entire database is encrypted using a single key due to which it is difficult to define access policies [1] [2] [9] [10]. The assumption is only limited to protecting data from the Cloud Service Provider, while the users have full access to the database [3]. However, in a real world scenario complete access to the whole database is not acceptable. It is also difficult to assign access privileges to each user, as well as there is a possibility of collusion attack by the revoked user and the CSP. That is why, it is to be ensured that a user has selective access to the database [4] [7].

Column-based encryption seems to be impractical as it is not possible to assign access privileges to users. So, one has to rely on third party applications or cumbersome stored procedures, which makes it impossible to provide information to the user instantly [3]. Some of the researchers used row-based encryption for ensuring security as it is easy to assign access privileges to users, based on the distribution of decryption keys. Francesco [3] scheme relies on main memory for storage and row-based encryption is used for implementing security. In this model, there are 'n' nodes (a trusted entity) one for each user and a synchronizer (an untrusted entity) who synchronizes the data stored at these nodes. Problem associated with the above approach is that it is costly and difficult to maintain and synchronize such a big number of nodes. Security is also at risk as all the keys are with the synchronizer.

Instead of these issues we preferred row-based encryption over others, since it strengthen security as it is impossible to gather all the encryption keys. This is one of the reasons why row-based encryption is preferred for transactional databases. In this paper, the proposed scheme presents a modified row-based encryption technique, in which some changes at the Database Owner and Cloud Service Provider are done to make it sure that all the constraints and triggers are applicable.

III. MODEL AND ASSUMPTIONS

The model is composed of a Database Owner, Cloud Service Provider and users associated with the Database Owner [2] [3] [4]. In the proposed model, neither the DO, nor the user has to be always on-line, but CSP is always on-line as it is done in [11]. Initially, user registers itself with the DO, who in turn provides encryption key and token to the user. The proposed scheme assumes that the algorithm used for generation of encryption keys is secure at DO. DO come on-line when a new user is to be registered. DO deliver this database to the CSP after encrypting all the rows using the corresponding encryption keys.

Symbol Table	
DO	Database Owner
CSP	Cloud Service Provider
PU	Public Key
PK	Private Key
EK	Encryption
DK	Decryption
PUCSP	Public Key of CSP
PRCSP	Private Key of CSP
PUDO	Public Key of DO
PRDO	Private Key of DO
PUUSR	Public Key of user
PRUSR	Private Key of user
UID	User Identity
D-H	Diffie-Hellman
$X_{A/B}$	Chosen Secret Key
$Y_{A/B}$	Calculated Public Key
K_S	Secret Session Key
q	Prime number
p	Primitive root

Now authentic user's requests are directed to the CSP, who in turn process user queries in a confidential manner. This scheme assumes that no outsider can breach CSP's security. Communication between CSP and users is made secure using a modified Diffie-Hellman Key Exchange and Public Key Encryption. D-H Key exchange is chosen, as it generates secret session keys each time the session expires. With the help of the session key, proposed scheme achieves secure data exchange as a unique key is exchanged each time for every new session.

IV. PROPOSED SCHEME

The proposed approach consists of a modified row-based encryption technique which helps us to achieve secure and efficient data access control. The proposed approach also uses client side storage, and selective encryption to increase performance of the system. In this section pseudo code of the algorithms and the notations used in it are also provided. An example of the proposed scheme that shows resemblance of it with the real life scenarios is also described in Figure 2.

The traditional database outsourcing model is shown in Fig1, where whole database is encrypted using a single key, and that key is distributed to the users. The assumption is only limited to protecting data from the Cloud Service Provider, while the users have full access to the database [3]. However, in a real world scenario complete access to the whole database is not acceptable. In this scenario usually over-encryption [17] or token-based authentication is used to handle revocation. Instead of these security primitives there is a possibility of collusion attack in which a malicious user passes the key to the CSP to attain full access to the database. To avoid this DO has to re-encrypt the whole database with a new key which is not possible in a real scenario. So, there is a need to implement a mechanism which can allow a controlled and secure access to the outsourced databases.

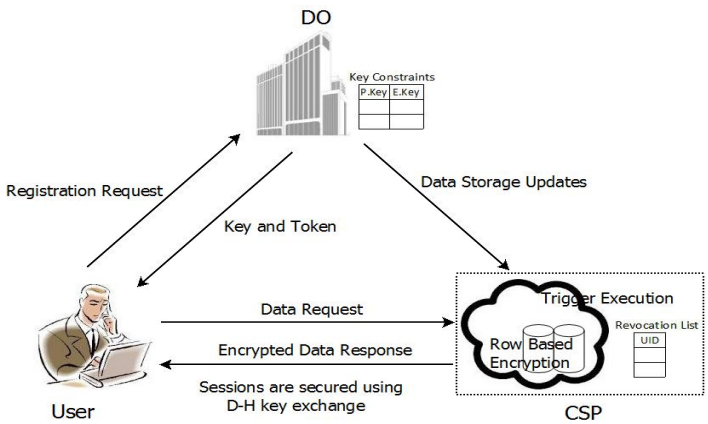

Fig.2. Our Proposed Database Outsourcing Model.

The user details which are sent to the DO and the results of the queries which is obtained from CSP must be secured in transit, to counter any security threats from the malicious outsiders. The proposed scheme will explain how to achieve the required security standards and efficient access control.

The proposed scheme is illustrated by the example shown in Fig 2. Here, the owner can be a bank who uploads his account holder's details on to the CSP server and the user is a customer (account holder) holding an account in the bank who access his account details from the CSP server. Whenever a new user opens an account in the bank, an entry is created for the user containing his personal and account details. Confidential details (which can reveal the identity of the account holder) are encrypted using a unique key, and the rest are kept as it is. Then the resulted entry is delivered to the CSP. Bank stores Primary key and Encryption key corresponding to the user in the table maintained at the bank side. At last, Bank delivers a packet containing UID, Account number, encryption key and a token to the user. This information is used by the user later on to retrieve it's account details from the CSP server. The proposed approach uses MD5 algorithm for message integrity and session keys (generated with the help of D-H key exchange algorithm) for confidentiality.

The procedure that the CSP uses after receiving data tables and revocation list from the DO is illustrated in algorithm1. CSP decrypts the message using its own private key and the public key of DO and stores the encrypted data tables and revocation list in its storage. Since, data tables are encrypted using row based encryption technique and the keys with which the rows are encrypted are unknown to the CSP. So, CSP will not be able to know the actual data. Revocation list received by the CSP contains UID of the users which are revoked. So, whenever a request of a user is encountered, his UID is matched with the revocation list. Users request is rejected, if UID of the user is there in the list.

Algorithm 1 Steps that CSP follows after receiving data table and revocation list from DO

Step 1: Storing encrypted database tables and revocation list

$ResultSet_0 \leftarrow Receive(EK_{PUCSP}(EK_{PRDO}(RevoList))$;

$ResultSet_i \leftarrow Receive(EK_{PUCSP}(EK_{PRDO}(ET_i))$;

Step 2: Updating the Database

$T_i \leftarrow DK_{PRCSP}(DK_{PUDO}(ResultSet_i))$;

Step 3: Updating Revocation List

$RevoList \leftarrow DK_{PRCSP}(DK_{PUDO}(ResultSet_0))$;

The process of acquisition of the keys is described in the algorithm 2. In it the user asks Database Owner for the keys which he will use for further communication with the Database Owner and Service Provider. In response Database Owner send public/private keys and a symmetric key, with which the Database Owner encrypts the row which contains user details.

Algorithm 2 Key acquisition phase.

Step 1: User requests the DO for the keys.

$Send(EK_{PUDO}(EK_{K_{ST}}(N1||req)||K_{ST}))$;

Step 2: DO responds with public/private keys and a symmetric key, with which they DO encrypts the row which contains user details.

$Receive(EK_{K_{ST}}(EK_{PRDO}(N1||req||K||PRUSR||PUUSR)))$;

The procedure required when a new user is to be added is described in algorithm 2. New user needs to send a registration request to the DO. DO creates an entry for the user containing his details and then generates a key using an algorithm. DO encrypts the entry with that key. Then DO stores Primary key and Encryption key corresponding to the user in the table maintained at its end and delivers that encrypted entry to the CSP (entry is first encrypted with its private key and then by public key of the CSP for the purpose of authentication and confidentiality between CSP and DO). At last, DO delivers a packet containing UID, encryption key and a token to the user. This information is used by the user later on to retrieve his details from the CSP server. The nonce and timestamps in the request and reply message serve the purpose of replay and man-in-the-middle attack avoidance.

Algorithm 3 Algorithm for registration of a new user

Step 1: User sends an encrypted registration request to DO
$\text{Send}(EK_{PUDO}(EK_{PRUSR}(\text{UserDetails})))$;

Step 2: DO updates Key Constraints table at its end
KeyTable ← add(UID, EK_{Ui});

Step 3: DO encrypts the row and sends it to the CSP
$\text{Send}(EK_{PUCSP}(EK_{PRDO}(EK_{Ui}(row_{Ui}))))$;

Step 4: CSP Updates its copy of the database
$DB_{CSP} \leftarrow \text{insert}(DK_{PUCSP}(DK_{PRDO}(EK_{Ui}(row_{Ui}))))$;

Step 5: Now the user can directly contact CSP for retrieving his details.

After the keys and token are delivered to the user, it can request CSP for the details. If UID of the user is not present in the revocation list, D-H is initiated by CSP. This fulfills the design objective of not keeping the DO always online. Algorithm 3 describes the use of Modified D-H key exchange algorithm to get a shared session key K_S for the purpose of secure communication between CSP and user. The proposed scheme uses Modified D-H key exchange to prevent man-in- the-middle attack by encrypting the Diffie-Hellman parameters and using nonce in each direction.

CSP encrypts the query result (r_i) and it's digest (D_i) using the shared session key K_S which is generated from the Modified D-H key exchange. This kind of encryption ensures confidentiality of the message between CSP and user as no one is able to read the message except the user. Session key remains valid for a predefined period of time which ensures secure communication over a period of time. The user decrypts the message upon receiving an encrypted response. After decryption user finally have access to information related to him in a secure and efficient manner.

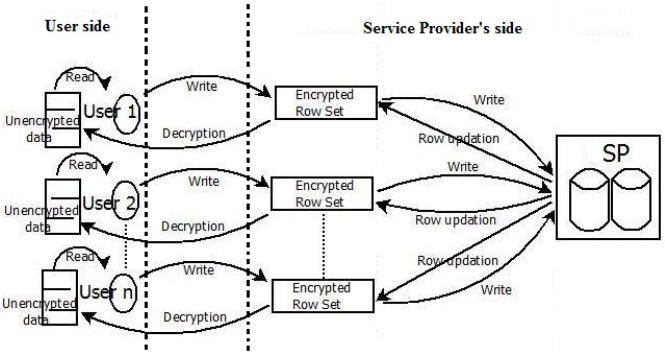

Fig.3. Read and write operations in our model.

The execution process of read and write operations is shown in Figure 3. In the proposed approach, cached row sets are used instead of result sets, as cached row sets doesn't require a database connection. As soon as user submits his login details, an encrypted RowSet is created, which resides with the instance of the server process with which the user is communicating. Whenever read operation is performed by the user, details are read from cached row set only, and whatever changes he has made are updated to the database residing with the Service Provider. RowSet is refreshed after a regular interval of time. This approach helps in improving concurrency, and maintaining consistency at the database.

Algorithm 4 Algorithm for secure communication between CSP and user

Step 1: User sends request for data access to CSP
Send (UID, Token, q, N_1);

Step 2: CSP checks revocation list if the user is not in the list then only query is processed
StorageArray← Receive(UID, Token, q);
if (StorageArray[1] = = RevoList(UID)) **then**
Goto step 7
else
Goto step 3
end if

Step 3: CSP and user exchanges a session key K_S generated with the help of modified D-H Key Exchange Algorithm.

Step 4: CSP encrypts the data using shared session key
$EO_i \leftarrow EK_S(EK_{Ui}(row_{Ui}))$;

Step 5: CSP sends the encrypted data to the User
Send (EO_i, $N_1 + 1$);

Step 6: Exit;

Step 7: CSP sends a rejection message to the User
Send (request cannot be granted);

V. SECURITY AND PERFORMANCE ANALYSIS

A. Security Analysis

In this section, various cryptographic primitives are analyzed in terms of their scalability and strength.

1) Data confidentiality: In database outsourcing, data confidentiality is a major issue because data of the user is sensitive and CSP is not in trusted domain of Database owner. So, the database cannot be stored in unencrypted form at CSP. The proposed scheme, uses row-based encryption, where each row is encrypted using a unique key. Since, size of the row is too small and cipher text of it after encryption is also small. So, this kind of encryption is very much comparable to one time pad. In cryptology, one time pad (OTP) encryption has been proven to be impossible to crack. So, use of this kind of encryption makes this model unbreakable which helps us to preserve the data confidentiality of user data in cloud computing.

2) Authentication: Row-based encryption itself ensures authentication of the user as row containing the user details and details are encrypted with the user's key. Hence, row will be accessible to a particular user only. For handling revocation, the proposed scheme uses tokens which are provided to the user at the time of registration.

3) Integrity: In the proposed scheme, MD5 hash algorithm and modified Diffie-Hellman key exchange has been used to secure data integrity. Modified D-H key exchange algorithm is used to generate different keys for different sessions. CSP encrypts data using the session key to secure communication between user and CSP.

B. Performance Analysis

In terms of performance, the proposed scheme saves a lot of computation cost in comparison to other encryption techniques such as Over-encryption technique. In the scheme proposed rows of the database are encrypted rather than the whole database. So it is quite easy to process the SQL queries, and there is no need of full database re-encryption if user rights are revoked. The scheme is analyzed in terms of applicability, computation cost, concurrency etc. and the results are described below.

1) Applicability: Bennani [3] scheme uses RSA encryption multiple times, which requires a lot of computation work to be performed by the client device. This scheme is not applicable on mobile devices as mobile devices have low computation capability. However, the proposed scheme uses AES encryption which requires less computation, and it is applied once. Hence, uses of AES encryption increase the applicability of the proposed scheme.

2) Cost efficient: In cloud computing cost depends on the number of queries and computational steps. In Bennani [3] scheme encryption and decryption is being done multiple times at the CSP side due to which computations are increased. Hence, cost is increased. In the proposed scheme, CSP just has to execute a SQL query which results the reduction in cost.

3) Concurrency: Proposed scheme is highly concurrent as whenever user will login, row containing user detail will be delivered to him in encrypted form. Now, user can perform all type of operation on this row such as read and write operation. This scheme also preserves the changes user has made on row, and it is delivered to the outsourced database before he logs out.

VI. SIMULATION AND RESULTS

The proposed approach is implemented using Java Socket Programming technique and MS Access 2005. For encryption and generation of the keys, cryptographic API's javax.crypto and java.security are used. Separate processes for CSP server, DO server and User are created. Java Socket Programming is used to create client and server processes. RowSet class is used for transfer of encrypted row because it can be communicated and manipulated without the need of database connection, which is one of the basic requirements of the proposed scheme.

The proposed scheme is simulated as defined in section IV. Later on observable results are compared with scheme [3] as mentioned in related work. Scheme [2] is also simulated, but it was taking a lot of time as it requires more than one encryption and over encryption steps. So, it can't be compared with the schemes on the graph. That is the reason why results of scheme [2] are not shown in the graph.

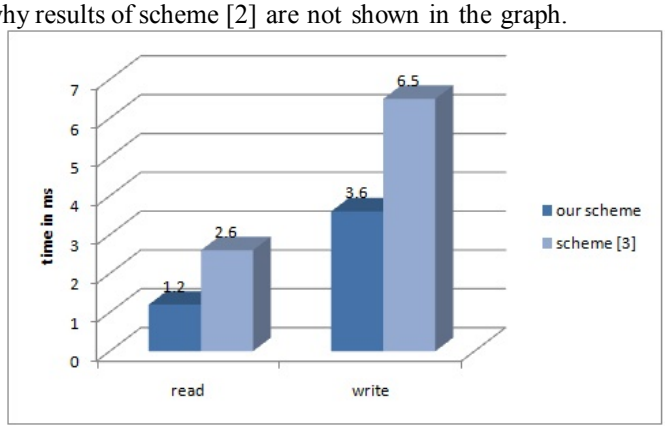

Fig.4. Comparison of operations with scheme [3].

As shown in figure 4, the proposed scheme performs quite well as compared to scheme [3]. For write operation, only three nodes are used for synchronization in scheme [3]. If, number of nodes will increase, time complexity will also increase in the scheme [3] and the proposed scheme will reflect more improvement in comparison to it.

VII. CONCLUSION AND FUTURE WORK

In this paper, the proposed scheme uses a modified row- based encryption technique for better access control and security. The proposed approach also uses client side memory and encryption to increase performance of the system. Selective encryption allows to execute triggers on the database, which is one of the issues when row-based encryption is used. The proposed scheme is very simple, secure and less time consuming in comparison to over-encryption technique or the one in which whole database is encrypted using a single key. It is flexible to use of any encryption technique for row-based encryption. In the proposed scheme, AES is used which proved itself better in comparison to RSA and other cryptographic algorithms when plaintext size is small.

In the proposed scheme, whole security is dependent on the key generation algorithm. So, there is a need of securing this algorithm from malicious outsiders. Instead of all these security primitives there is a possibility of information leakage from access patterns. The proposed work need to be extended, in such a way that the proposed model will be secured in every perspective. The proposed scheme is to be tested in a real cloud environment and with a large population of users.

REFERENCES

[1] H. Hacigumus; B. Iyer; S. Mehrotra; , "Providing database as a service," Data Engineering, 2002. Proceedings. *18th International Conference on, vol., no., pp.29-38*, 2002.

[2] N. Bennani; E. Damiani; S. Cimato; , "Toward Cloud-Based Key Management for Outsourced Databases," *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual, vol., no., pp.232-236*, 19-23 July 2010.

[3] F. Pagano; D. Pagano; , "Using in-memory encrypted databases on the cloud," *Securing Services on the Cloud (IWSSC), 2011 1st International Workshop on, vol., no., pp.30-37*, 6-8 Sept. 2011.

[4] S. Sanka; C. Hota; M. Rajarajan; , "Secure data access in cloud computing," *Internet Multimedia Services Architecture and Application(IMSAA), 2010 IEEE 4th International Conference on, vol., no., pp.1- 6*, 15-17 Dec. 2010.

[5] C. Curino; E. Jones; R. A. Popa; N. Malviya; E. Wu; S. Madden; H. Balakrishnan; N. Zeldovich; , "Relational Cloud: A Database Service for the Cloud," *Innovative Data Systems Research, 2011 5th Biennial Conference on, vol., no., pp.*, Jan. 2011.

[6] J. Heurix; T. Neubauer; , "On the Security of Outsourced and Untrusted Databases," *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on, vol., no., pp.125-132*, 18-20 Aug. 2010.

[7] Wang Xiaoming; Zhang Yanhui; , "A Dynamic Access Control Scheme for Outsourced Database," *Network Computing and Information Security (NCIS), 2011 International Conference on, vol.1, no., pp.3-7,* 14-15 May 2011.

[8] Wang Zheng-Fei; Ai-Guo Tang; , "Implementation of encrypted data for outsourced database," *Computational Intelligence and Natural Computing Proceedings (CINC), 2010 Second International Conference on, vol.2, no., pp.150-153,* 13-14 Sept. 2010.

[9] H. Hacigumus; B. Iyer; S. Mehrotra; C. Li; , "Executing SQL over encrypted data in the database-service-provider model," *In Proc. of the ACM SIGMOD 2002, Madison, WI, USA,* June 2002.

[10] R. Agrawal; J. Kierman; R. Srikant; Y. Xu; , "Order preserving encryption for numeric data," *In Proc. of ACM SIGMOD 2004, Paris, France*, June 2004.

[11] S. Yu; C. Wang; K. Ren; W. Lou; , "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," *in Proc. of IEEE INFOCOM*, 2010.

[12] E. Mykletun; M. Narasimha; G. Tsudik; , "Authentication and integrity in outsourced databases," *in Proc. of the ISOC Symposium on Network and Distributed Systems Security*, 2004.

[13] E. Damiani; S.D.C Vimercati; S. Jajodia; S. Paraboschi; P. Samarati; , "Balancing confidentiality and efficiency in untrusted relational dbmss," *Proceedings of the 10th ACM conference on Computer and communications security, pages 93102*, 2003.

[14] A. Zych; M. Petkovi; W. Jonker; , "Efficient key management for cryptographically enforced access control," *Comput. Stand. Interfaces, 30(6):410417*, 2008.

[15] J. He; M. Wang; , "Encryption in relational database management systems," *In Proc. Fourteenth Annual IFIP WG 11.3 Working Conference on Database Security (DBSec00), Schoorl, The Netherlands*, 2000.

[16] R. Agrawal; A. Evfimievski; R. Srikant; , "Information sharing across private databases," *in Proc. of the ACM SIGMOD Conf., 2003, pp. 8697*, 2003

[17] S.D.C. Vimercati; S. Foresti; S. Jajodia; S. Paraboschi; P. Samarati; , "Over-encryption: management of access control evolution on outsourced data," *In VLDB 07: Proceedings of the 33rd international conference on Very large data bases, pages 123134. VLDB Endowment*, 2007.

[18] H. Hacigumus; S. Mehrotra; , "Performance-conscious key management in encrypted databases," *Research Directions In Data And Applications Security XVIII: IFIP TC 11/WG 11.3 Eighteenth Annual Conference On Data And Applications Security*, July 25-28, 2004, Sitges, Catalonia, Spain, 2004.

[19] E. Damiani; S.D.C. Vimercati; S. Foresti; S. Jajodia; S. Paraboschi; P. Samarati; , "Selective data encryption in outsourced dynamic environ- ments," *Electronic Notes in Theoretical Computer Science, 168:127142*, 2007

[20] W. Wang; Z. Li; R. Owens; B. Bhargava; , " Secure and efficient access to outsourced data," *In CCSW 09: Proceedings of the 2009 ACM workshop on Cloud computing security, pages 5566,* New York, NY, USA, 2009. ACM.

[21] M. Kantarcioglu; C. Clifton; , "Security issues in querying encrypted data," *in Proc. of the IFIP Conference on Database and Applications Security*, 2005.

[22] R. Sion; , "Secure data outsourcing," *in Proc. of the VLDB Conf., 2007,pp. 1431-1432*, 2007