International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

Survey on Schema for Concurrent Access over Encrypted Cloud Database Using SQL Operations for Access Control Mechanism

Basavraj S Kamlapure
Savitribai Phule Pune University, DYPIET, Ambi, Pune, India
basavrajkamlapure@gmail.com
Neha B Bhosle
Savitribai Phule Pune University, DYPIET, Ambi, Pune, India
neha.bhosle92@gmail.com
Amol B Thange
Savitribai Phule Pune University, DYPIET, Ambi, Pune, India
amolthange6@gmail.com
Sonal B Hinge
Savitribai Phule Pune University, DYPIET, Ambi, Pune, India
soniahinge@rediffmaill.com

Abstract
The key issues with respect to cloud administrations are security, accessibility, adaptability and data classification. Security, versatility and accessibility of any cloud administrations are taken cared by the cloud suppliers, yet usage of data classification is open issue for the occupants. The proposed design incorporates data encryption, key administration, data decryption and all issues identified with cloud administrations. This design utilizes SQL operations over cloud data. Formal models portray the proposed answer for implementing access control and for ensuring privacy of data and metadata.

Keywords: Encryption; Decryption; Access control; Database; Confidentiality

Nomenclature

MuteDB- Multi-User relational Encrypted Database, DBA- Database Administrator, KDC- Key Distribution Center, ABE- Attribute Based Encryption, ABS- Attribute Based Signature, KAC- Key-Aggregate Cryptosystem, TPA- Third Party Auditor, PL- Priority List, MAC- Mandatory Access Control, DAC- Discretionary Access Control.

1. INTRODUCTION

In cloud setting, the basic data in cloud experiences issues like data security, versatility and classification. The explanation for dissemination of cloud database administrations is classification of the data that client used to store on the cloud. We expect to accomplish detachment and privacy of the data. Existing frameworks offers separate answers for data classification and segregation. As some of frameworks supports SQL operations on encrypted cloud data yet leaves access control to the cloud supplier or middle of the road cloud servers. What's more, some of frameworks supports access control without inclusion of cloud supplier's however does not permit performing SQL operations on encrypted cloud data.

The proposed architecture named as MuteDB, assurances the data confidentiality of encrypted cloud data by utilizing SQL operations and authorizes access control policies for multiuser by utilizing some specific techniques. This is the main architecture that joins both the arrangements. The proposed architecture is intended for cloud administration situations where numerous clients can access cloud benefits perhaps from various topographical areas. This architecture works in element situations that implies, clients and access controls can be changed when required, without restoring or redistributing client credentials. This system does not use any intermediate server which can become a point of failure.
The MuteDB's execution is assessed through a model that subject to various question workloads in view of standard and as of late proposed database benchmarks. A trial result demonstrates that, this architecture does not influence the adaptability of unique cloud administration.
MuteDB permits the endeavors to utilize cloud database administrations with the confirmation of ensuring data confidentiality and versatility of administrations.

2. BACKGROUND

Possibly there exist four roles in this architecture: The tenant Database administrator (DBA), the tenant database users, the cloud provider employees and external users.
The DBA assumes part of circulating access controls to clients of occupant association. He is in charge of introducing and designing the database which actualizes access control policies and overseeing client credentials.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

Outside assailants don't have entry to the foundation and data of inhabitant association nor to the cloud suppliers. They can perform diverse assaults to access occupant's data.
The cloud insiders are representatives of the cloud supplier. They are in charge of facilitating the database administration of occupant association. Their conduct is straightforward, that is, they might be occupied with accessing the inhabitant's data yet they don't abuse the data.
Inhabitant database clients have entry to data put away in the cloud. Each client is limited to access to the data as credentials relegated by DBA. The bit of accessible data is characterized by the access control policies of the inhabitant association.
By utilizing access control policies, the operations performed by clients of inhabitant association can be controlled. Within and outer aggressors in cloud that have ruptured the cloud server, can't access the private data, on the grounds that MuteDB encodes data with SQL-mindful encryption calculation and cloud supplier never acquires the decryption key.

3. LITERATURE REVIEW

3.1. Distributed, Concurrent and Independent Access to Encrypted Cloud databases [1]:

In the context of cloud, the classified data is taken care of by untrusted outsider. In this situation, data is overseen in two ways: occupant's plain data is made accessible to just trusted gathering, In untrusted connection, the data is encrypted. Every one of these operations are performed by untrusted outsiders and data is put away on cloud. This raises a bottleneck issue at untrusted outsider when number of customers solicitations to access data. So to conquer this issue, another architecture called as SecureDBaaS is outlined where moderate untrusted outsiders are wiped out. SecureDBaaS furnishes cloud database administrations with data confidentiality furthermore adds highlight that permits to executing simultaneous operations on encrypted cloud data. This architecture permits topographically circulated customers to access cloud data straightforwardly and simultaneously adjust the encrypted data by utilizing SQL explanations. Dispensing with middle of the road intermediary servers permits SecureDBaaS to accomplish accessibility, unwavering quality and versatility of cloud DBaaS.[1]

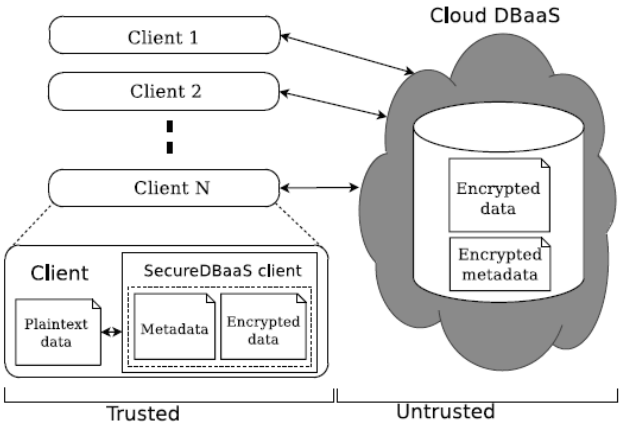

Fig1. SecureDBaaS architecture [1]

Fig1 describes the overall architecture of SecureDBaaS.
This architecture down not contains any transitional intermediary which unravels single point failure of intermediary server based architecture. The data in this architecture contains plaintext data, encrypted data, metadata and encrypted metadata. This architecture utilizes distinctive approach to store data and metadata in the cloud databases. The customer utilizes SQL explanations to recover data and metadata from untrusted cloud database. Numerous customers can access this data simultaneously and freely.
The occupant's data is put away in relational database and Encrypted inhabitant data is put away in secure tables into the cloud database. SecureDBaaS takes after validation and approval process gave by unique DBMS server. After fruitful confirmation, by utilizing SecureDBaaS, the client communicates with cloud database. SecureDBaaS distinguishes the operations that client is performing and recognizes the tables included and recovers the metadata from the cloud database. The encrypted metadata is decrypted utilizing expert key and afterward by utilizing their data, the first SQL is deciphered into an inquiry that can be executed on encrypted data. The consequence of deciphered question contains encrypted inhabitant data and metadata. This outcome is gotten by SecureDBaaS and decrypted and conveyed to the client.

3.2. Decentralized Access Control With Anonymous Authentication of Data Stored in Clouds [2]:

This schema is decentralized in nature, in which disseminated access control of data stored in the cloud and just substantial clients can access them. It verifies the clients who are going to access data in the cloud. Amid verification, the character of client is shielded from the cloud. This is a decentralized architecture where a few quantities of KDC's are accessible for key administration. The validation and access control are impact safe, so that no two clients can plot and access data. This plan is flexible to replay assaults, so that an author, whose traits and keys have been repudiated, can't compose back stale data. This convention permits different read and composes operations on data stored in the cloud and the costly operations are finished by the cloud.

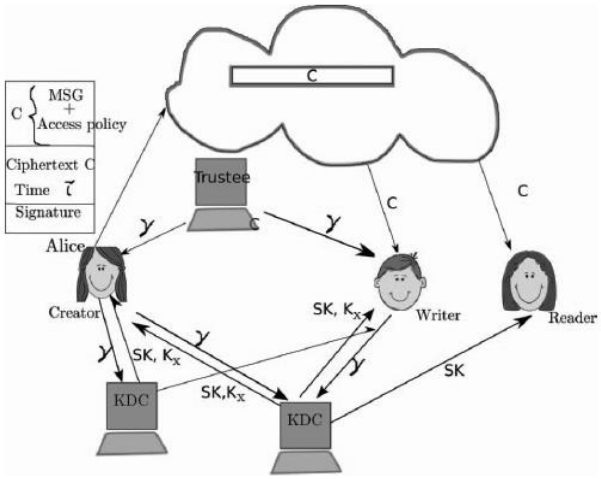

Fig2. Cloud model [2]

According to this scheme, a user can create a file and store it on cloud securely. This scheme consists of two protocols: ABS and ABE.
As appeared in above Fig2, there three clients, a creator, reader and writer. In the first place creator presents id to the trustee and gets a token from the trustee. There are a few KDC's available in the scheme, creator shows his token to one of the KDC and gets a secret key for encryption/decryption process or for marking into the framework. The client's data is encrypted under the access strategy. This access arrangement chooses who can access the data stored on the cloud. The creator settles on a case strategy to demonstrate his validness and signs the message under case. The cloud checks the mark and stores this ciphertext. At the point when a reader needs to peruse the message, cloud checks for whether asking for reader has characteristics coordinating with the access policies or not. In the event that yes, then reader can decode the first message and get.
At the point when a client solicitations to cloud for data, the cloud sends ciphertext utilizing SSH convention. At that point by utilizing ABE calculation decryption is finished.
To write in effectively existing document, the client must send his message with case policies as done amid record creation. The cloud then checks the policies and if client is bona fide, then that client is permitted compose data on that document.
There is stand out confinement with this scheme is that; the cloud knows the access strategy of every record stored in the cloud.

3.3. Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage [3]:

Key-Aggregate Cryptosystem scheme is for safely, productively and adaptably sharing data with others in the cloud.
This scheme produces steady size ciphertexts and assignments of decryption rights for any arrangement of ciphertexts are conceivable. In this scheme, a secret key holder can discharge steady size total key for decisions of ciphertexts in cloud and different ciphertexts stay confidential.
In this scheme, client encodes data utilizing an open key furthermore by utilizing identifier of ciphertext which is called as class. These ciphertexts are further arranged into various classes. The key owner can separate secret keys for various classes by utilizing a master-secret key. General society key, master-secret key, total key and ciphertext utilized as a part of this scheme are all of consistent size.
This scheme comprises of five polynomial-time calculations:
Setup: This is finished by owner of data to setup a record on an untrusted server. A security level parameter and number of ciphertext classes are given as info. It yields an open framework parameter.
KeyGen: This is finished by owner of data to produce an open/master-secret key pair.
Encode: This is finished by client who need to scramble the data. The first message, open key, ciphertext class are given as information and produces ciphertext as yield.
Remove: This is finished by owner of the data for assigning the unscrambling power for an arrangement of ciphertext classes. Master-secret key and set of files of ciphertext classer are given as information. It yields total key.
Unscramble: This is finished by the client who got a total key which is produced by Extract.
This scheme permits the owner of data to share their data in confidential and specific route, with settled and little ciphertext development and by conveying a little or single total key to each approved client.

As appeared in above Fig3, when a client needs to share data on the server, User first performs setup and gets open framework parameter and executes KeyGen to get open/master-secret key pair. At that point by utilizing these keys, client can scramble the first data and store it on distributed storage

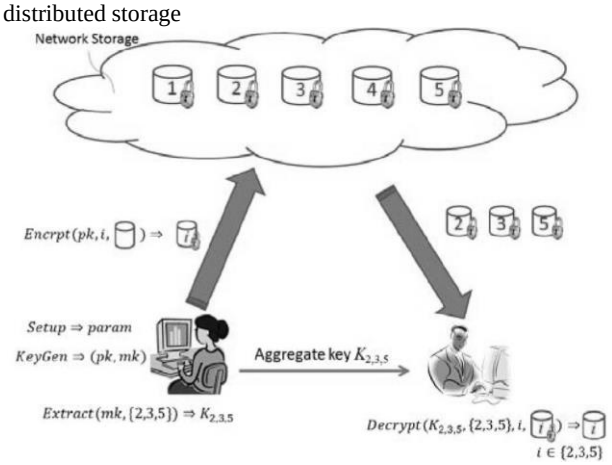

Fig3. Data sharing in cloud using KAC [3]

When user wants to share the data stored in the cloud with others, user performs Extract and computes aggregate key. This key can be given to other user by using secure e-mail.
User after obtaining aggregate key can download the encrypted data from cloud and by using aggregate key can decrypt the encrypted data.

3.4. Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data [4]:

It vital to empower seek benefit and investigate privacy preserving on encrypted cloud data. As there are bigger number of clients and archives on the cloud, it is required to permit multi-keywords look in the hunt administration. This scheme characterizes the main architecture which permits multi-keyword seek over the encrypted cloud data. This scheme is exceptionally alluring in distributed storage technology as this positioned look framework empowers client to discover most important data rapidly and wipe out superfluous system activity. This scheme enhances the output precision and upgrades client seeking knowledge by supporting multiple keywords inquiry.
This scheme depends on productive similitude measure of "Coordinate matching" i.e. however many matches as could reasonably be expected, to catch importance of data to the inquiry question.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

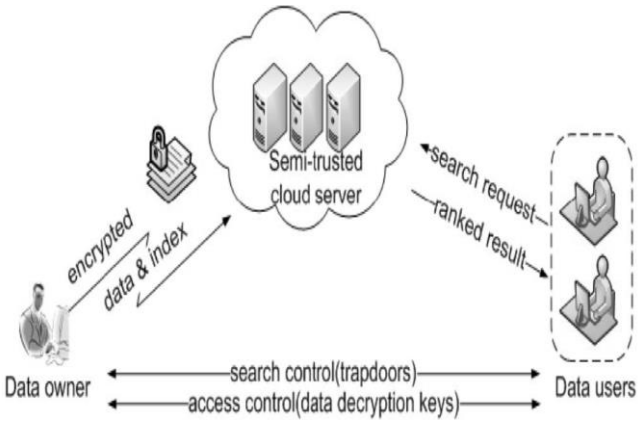

Fig4.Architecture of search over encrypted cloud data [4]

As shown in the above fig4, there are three distinct elements: the data owner, data user and cloud server. The data owner is a user who has gathering of data records which is stored on the cloud server in the encrypted structure. To empower productive inquiry administration over encrypted data, the data owner will manufacture encrypted seek list and after that stores both data with its encrypted look record over cloud. To scan a data archive for given pursuit keyword, an authorized user procures a trapdoor through inquiry control system. Cloud server gets this trapdoor by a user, and cloud server is capable to look the list and returns the arrangement of encrypted archives. To enhance this record recovery exactness, the query item is positioned by cloud server as indicated by some positioning criteria.
Framework:
With spotlight on list and inquiry, this framework comprises of four calculations as takes after:
Setup: It takes security parameter as info and yields a symmetric key by data owner.
BuildIndex: According to the data set, data owner form the searchable file, and encrypted by symmetric key and after that stored on the cloud server.
Trapdoor: This calculation produces trapdoor for keywords utilized as a part of looking.
Question: Cloud server gets a hunt inquiry, it performs a positioned seek on record with the assistance of trapdoor and returns positioned id rundown of top archives.

3.5. Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud [5]: Cloud gives storage where users can store their data and access it from anyplace. This stored data can be shared by numerous users. Be that as it may, it is essential to keep up open auditing for such shared data. Oruta is first privacy-preserving instrument that permits open auditing on shared data storage in the cloud. The trustworthiness of data in distributed storage is important as data stored on untrusted cloud can be lost or tainted. To secure respectability of cloud data, open auditing is performed by presenting an outsider auditor (TPA). An exceptional issue during the time spent auditing for shared data in cloud is the manner by which to protect character privacy from the TPA, in light of the fact that the personalities of endorsers on shared data may demonstrate that a specific user in the gathering is a higher significant focus than others. The data of characters of such users is confidential and it ought not be uncovered to whatever other outsider. In Oruta, it uses ring marks to build homomorphic authenticators. In this scheme, TPA can check the honesty of shared data for a gathering of users without recovering whole data. The personality of underwriter in shared data is kept private from the TPA. This scheme likewise supports bunch auditing which can audit multiple mutual data at the same time in a solitary auditing assignment. This scheme utilizes irregular concealing to bolster data privacy amid open auditing and influence record hash tables to bolster dynamic operations on shared data.

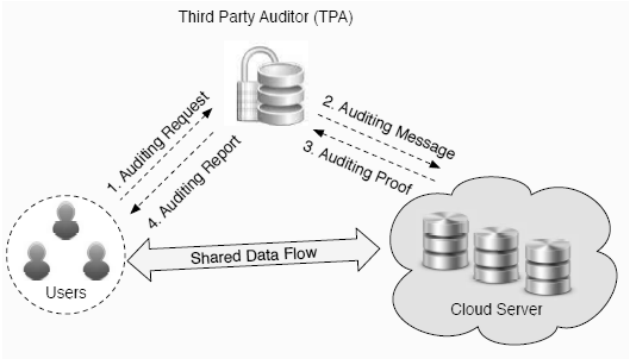

Fig5. System model [5]

As shown in Fig.5, this scheme includes three gatherings: The cloud server, outsider auditor (TPA) and users. Users can be unique user and number of gathering users. Unique user makes the data and stores it on the distributed storage and gathering users can access and change the common data. Cloud server holds data of shared data and its confirmation data. The TPA can confirm the trustworthiness of shared data in the cloud server.

This scheme just demonstrates to audit the uprightness of imparted data in cloud to static gatherings. Static gatherings are predefined before data is stored on the cloud and participation of gathering can't be changed amid data sharing. The first user can choose who can access the data before putting away on the cloud. At the point when unique user or any gathering user needs to check the trustworthiness of shared data, user will first send an auditing solicitation to the TPA. TPA creates an auditing message to cloud server subsequent to getting auditing demand and Cloud server will gives an auditing verification of shared data. At that point TPA confirms the rightness of the auditing proofs and sends an auditing report to the user according to the consequence of confirmation.

3.6. Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud [6]:

The cloud comprises of data storage and sharing administrations which gives the user simple adjustment and sharing of data as a gathering. The users in the gathering process marks on all pieces in the common data in order to guarantee honesty of that mutual data. As the adjustments in the mutual data are performed by various users, diverse squares in that common data are along these lines marked by various users. At the point when any user is denied from the gathering them the pieces which were beforehand marked by this user should be re-marked by the current user due to security reasons. the most uncomplicated and basic technique permits any current user to download any relating part of the mutual data and re-sign it amid the user disavowal, however this turns out to be wasteful in light of the

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 09
May 2016

huge size of the common data. This paper proposes another and extraordinary instrument of open auditing for the honesty of imparted data to productivity if there should arise an occurrence of user denial. The current user need not download and re-sign pieces amid user repudiation as we permit the cloud to-sign squares in the interest of the current user. Regardless of the fact that some a player in shared data is re-marked by the cloud, an open verifier can audit the honesty of that common data without downloading the whole data from the cloud.

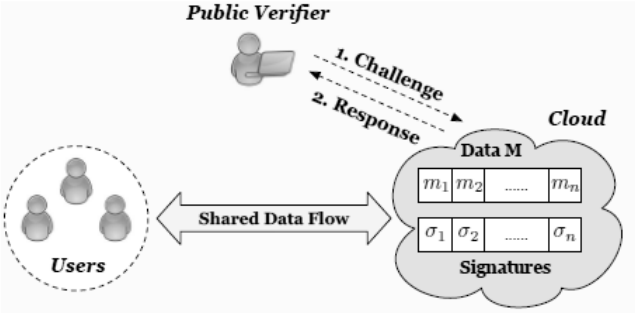

Fig6. Architecture of Panda [6]

This component comprises of six calculations: KeyGen, ReKey, Sign, ReSign, ProofGen, ProofVerify. The KeyGen permits the user in the gathering to produce his/her open key and private key. The ReKey empowers the cloud to figure a re-marking key for every pair of users in the gathering. At the point when the first user makes shared data in the cloud, he/she processes a signature on every piece as in Sign. After that, if a user in the gathering alters a piece in shared data, the signature on the adjusted square is additionally figured as in Sign. In ReSign, a user is renounced from the gathering, and the cloud re-signs the pieces, which were already marked by this denied user, with a leaving key. By means of the test and-reaction convention between the cloud and an open verifier the data trustworthiness of the common data is confirmed. The cloud can produce a proof of ownership of shared data in ProofGen under the test of an open verifier. In ProofVerify, an open verifier can check the rightness of a proof reacted by the cloud. In ReSign the cloud changes over signatures of the denied user into the signatures of the first user without loss of sweeping statement. Another method for choosing the re-marking key is by requesting that the first user make a need list (PL). Presently, the primary user appeared in the PL is chosen when the cloud needs to choose which existing user the signatures should be changed over to. To guarantee the rightness of this PL it is to be marked with the private key of the first user.

3.7. Role Based Access Control Backup and Restoration Ontology [7]:

Distributed computing is the most developing technology now days. The most vital is to safeguard the data and, privacy of user. Access control assumes an underlying part of permitting, denying and limiting access to framework. It can likewise distinguish users endeavoring to framework unauthorized. This framework serves security towards number of user per part, exchanges done by users furthermore adds highlight to take reinforcement and restore data. Just suitable user can give and disavow the validation.

Mandatory Access Control (MAC), Discretionary Access Control (DAC) were dangerous for distributed frameworks and dealing with the access to assets and framework was troublesome so new access model is presented known as Role Based Access Control Reference Ontology depicts a RBAC model utilizing a part ontology for Multi-Tenancy architecture for particular space. Multifaceted nature is decreased with the assistance of ontology

Three essential guidelines are characterized for RBAC:

Part task: A subject can allow authorization just if the subject has been doled out a part.

Part approval: Rule guarantees that users can tackle parts for which they are authorized.

Consent approval: Permission is authorized for the subject's dynamic part.

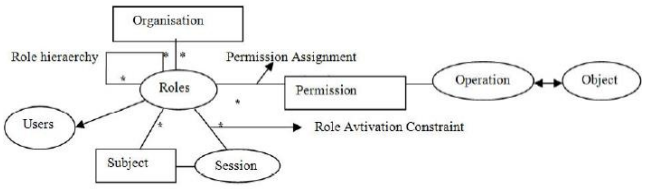

Fig7. Architecture of RBAC [7]

Aim of the framework is to save privacy of user. Ontology gives access to confinement and reflection from genuine framework.

For instance in figure we allude to a substance as a user and a data object as a record. Part is an arrangement of operations that user performs on data. Whenever operations and parts as for subject to authoritative policies, and operations cover, progressions of parts are built up. Access control assumes two parts validation and approval. While marking framework secret word is a case of confirmation and the person who can access record or not is approval. User can reassign or expel benefits powerfully without evolving consent. Multiple customers can store and oversee by the same programming. In any case, multi tenure emerges security issue because of sharing of programming. To defeat this issue RBAC works in two stages while allocating benefits. In first stage user appreciates rights by their parts. Also, in second stage progressively task and reclamation is finished.

4. CONCLUSION AND FUTURE SCOPE

Now a days security, accessibility, adaptability and confidentiality to a cloud data is fundamental. Every paper portrayed above assumes their parts however not each of the four. This framework beats all the above required parts furthermore gives access control at user side. Single point failure and bottleneck issues are stayed away from. Privacy limitations are not damaged Suitable for all SQL questions.

As the examination work begins on any framework and changes are done later to make it more gainful. The proposed framework is additionally an enhanced adaptation of the past existing frameworks.

5. ACKNOWLEDGEMENT

We would like to acknowledge and extend our heartfelt gratitude to our guide Prof. Sujay P. Pawar, DYPIET for his expert guidance and encouragement.

6. REFERENCES

[1] L. Ferretti, M. Colajanni, and M. Marchetti, "Distributed, concurrent, and independent access to encrypted cloud databases," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 2, pp. 437–446, 2014.

[2] Sushmita Ruj, Milos Stojmenovic and Amiya Nayak , "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds," IEEE Transactions on Parallel And Distributed Systems, VOL. 25, NO. 2, FEBRUARY 2014.

[3] Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, " Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage," IEEE Transactions On Parallel And Distributed Systems, VOL. 25, NO. 2, FEBRUARY 2014.

[4] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[5] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," in the Proceedings of IEEE Cloud 2012, 2012, pp. 295–302.

[6] Boyang Wang, Baochun Li and Hui Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud ," IEEE Transactions On XXXXXX, VOL. X, NO. X, XXXX 201X.

[7] D.F. Ferraiolo and D.R. Kuhn, "Role-Based Access Controls," Proc. 15th Nat'l Computer Security Conf., 1992.

[8] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Proc. 2010 IEEE Int'l Conf. Cloud Computing Technology and Science*, Nov.-Dec. 2010, pp. 693 – 702.

[9] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically scaling applications in the cloud," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 45–52, 2011.

[10] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Key management for multi-user encrypted databases," in *Proc. ACM Workshop Storage Security and Survivability*, Nov. 2005, pp. 74 – 83.

[11] L. Ferretti, M. Colajanni, and M. Marchetti, "Access control enforcement of query-aware encrypted cloud databases," in *Proc. Fifth IEEE Int'l Conf. on Cloud Computing Technology and Science*, Dec. 2013, pp. 717–722.

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Computer and communications security*, 2006, pp. 89–98.

[13] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: protecting confidentiality with encrypted query processing," in *Proc. 23rd ACM Symp. Operating Systems Principles*, Oct. 2011, pp. 85–100.

[14] E. Damiani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Key management for multi-user encrypted databases," in *Proc. ACM Workshop Storage Security and Survivability*, Nov. 2005, pp. 74 – 83.