iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** X **Month of publication:** October 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology(IJRASET)

A Secured frameworkfor SCAM (Scalable Access Control Mechanism) in Cloud Computing

Vaibhav Gandhi[#1], Prof. Prashant Lakkadwala[*2]

*M. Tech Student [#1],Assistant Professor#2, Computer Science & Engineering [#1*2]*

*Rajeev Gandhi Technical University,Bhopal[#1*2]*

Abstract:The paradigm that offers Cloud computing is advantages in economic aspects, by reducing flexible computing, capabilities limitless computing power and time to market. To use the full potential of cloud computing like transferring, processing and storing time of data by external cloud providers. To keep user data confidential from untrusted cloud servers, existing solutions use cryptographic methods and only disclose decryption keys to the authentic users. Unfortunately, these models are not applicable to cloud as the data owners and service providers are not in the same trusted domain. Therefore, Our proposed scheme enables the data owner to delegate tasks of data file creation ,encryption ,decryption ,re-encryption and user secret key update to cloud servers without disclosing data contents or user unique access structure information. Main issues such as privacy, scalability for key management, flexibility in access and user revocation which are the most important considerations for gaining scalability and flexibility. We achieve our design goals by a novel structuring , Advanced attribute based encryption in which a unique access structure is assign for each attributes In existing scheme revocation user details such as private key are updated manually after each user revocation. In our architecture at server side a ttp value (threshold value) is set, when it reaches the threshold value revoked users are updated and updating is performedby using atomic proxy cryptography technique of re-encryption for revocation of user to update the attributes of all the live users. This construction allows each data owner to access his data files with minimum online time and minimum overhead which the aim of our work.We formally prove the security of AABE based on security of the cipher text-policy attribute-based encryption (CP-ABE) scheme by Bethencourt et al. and analyze its performance and computational complexity.

Keywords: Cloud Computing, Data security, A-ABE(Advanced attribute based encryption),atomic proxy reencryption

1.INTRODUCTION

Cloud computing is the new trend of computing where readily available computing resources, are exposed as a service. These computing resources are generally offered as ‘pay-as-you-go’ plans and hence have become attractive to cost conscious customers [1]. Apart from the cost, cloud computing also supports the growing concerns of carbon emissions and environmental impact since the cloud advocates better management of resources [2]. We see a growing trend of off-loading the previously in-house service systems to the cloud, based primarily on the cost and the maintenance related burden. Such a move allows businesses to focus on their core competencies and not burden themselves with back office operations. As consumers move towards adopting such a Service-Oriented Architecture (SOA), the quality and reliability of the services become important aspects. In SOA terms, this agreement is referred to as a Service Level Agreement (SLA). This SLA serves as the foundation for the expected level of service between the consumer and the provider [2].

His tension makes sense as users want to maintain control of their data, but they also want to benefit from the rich services that application developers can provide using that data. So far, the cloud offers little platform-level support or standardization for user data protection beyond data encryption at rest, most likely because doing so is nontrivial [3].

Six important issues that must be addressed to ensure an organization or individual's use of cloud computing is not compromised.

- The first is “resource sharing”. On shared services, there is the possibility that another user on the same system may gain access inadvertently or deliberately to one's data, with potential for identity theft, fraud, or industrial sabotage.
- Second, because data is held offsite, data ownership might be compromised.
- Third, the intrinsic latency of transferring data back and forth for processing in the cloud means that some users might lower encryption levels to cut send and receive delays, giving rise to additional security concerns.
- Fourth, the issue of Service Line Agreements (SLAs) may lead to an organization being refused access to data and services if there are any technical, security or commercial disagreements between them and the cloud service provider.
- Fifth, data might be lost or otherwise compromised because of a technical or other failure on the part of the provider.

- Finally, negative aspects of interoperability and portability in which failure or attack of a virtual component in the processing and storage may compromise security.

II. CLOUD COMPUTING REVIEW

Attribute-Based Encryption (ABE) was first proposed by Sahai and Waters [1] with the name of Fuzzy Identity-Based Encryption, with the original goal of providing an error-tolerant identity-based encryption [2] scheme that uses biometric identities. In [3], Pirretti et al. proposed an efficient construction of ABE under the Random Oracle model and demonstrated its application in large-scale systems. Goyal et al. enhanced the original ABE scheme by embedding a monotone access structure into user secret key. The scheme proposed by Goyal et al. is called Key-Policy Attribute-Based Encryption (KP-ABE) [4], a variant of ABE. In the same work, Goyal et al. also proposed the concept of Ciphertext-Policy Attribute Based Encryption (CP-ABE) without presenting a concrete construction. CP-ABE is viewed as another variant of ABE in which cipher texts are associated with an access structure. Both KP-ABE and CP-ABE are able to enforce general access policies that can be described by a monotone access structure. In [5], Ostrovsky et al. proposed an enhanced KP-ABE scheme which supports non-monotone access structures. Chase [6] enhanced Sahai-Waters ABE scheme and Goyal et al. KP-ABE scheme by supporting multiple authority. Further enhancements to multi-authority ABE can be found. Bethencourt et al. [7] proposed the first CP-ABE construction with security under the Generic Group model. In [8], Cheung et al. presented a CCA-secure CP-ABE construction under the Decisional Bilinear Dife-Hellman (DBDH) assumption. In [8], the CCAsecure scheme just supports AND gates in the access structure. Towards proposing a provably secure CP-ABE scheme supporting general access structure, Goyal et al. [3] proposed a CP-ABE construction with an exponential complexity which can just be viewed as theoretic feasibility. For the same goal, Waters [4] proposed another CP-ABE scheme under various security assumptions. Aside from providing basic functionalities for ABE, there are also many works proposed to provide better security/privacy protection for ABE. These works include CP-ABE with hidden policy, ABE with user accountability [5], ABE with attribute hierarchy [6] and etc.

In cloud computing, there are different existing schemes that provide security, data confidentiality and access control. Users need to share sensitive objects with others based on the recipients ability to satisfy a policy in distributed systems. One of the encryption schemes is Attribute Based Encryption (ABE) which is a new paradigm where such policies are specified and cryptographically enforced in the encryption algorithm itself. Hence, the existing ABE schemes are of two types.Several schemes employing attribute-based encryption (ABE) have been proposed for access control of outsourced data in cloud computing; however, most of them suffer from inflexibility in implementing complex access control policies. Hierarchical attribute-set-based encryption (HASBE) by extending cipher text-policy attribute-set-based encryption (ASBE) with a hierarchical structure of users. This scheme not only achieves scalability due to its hierarchical structure, but also inherits flexibility and fine-grained access control in supporting compound attributes of ASBE. In addition, HASBE employs multiple value assignments for access expiration time to deal with user revocation more efficiently than existing schemes[24]. But this introduced heave computation overhead on data owner .then we remove this by introducing threshold value at server side for user revocation .as threshold level is achieved all the system must keys updated by re encryption process i.e atomic proxy re encryption .We formally prove the security of AABE based on security of the cipher text-policy attribute-based encryption scheme by Bethencourt et al.[23] and analyze its performance and computational overhead complexity.

Architecture

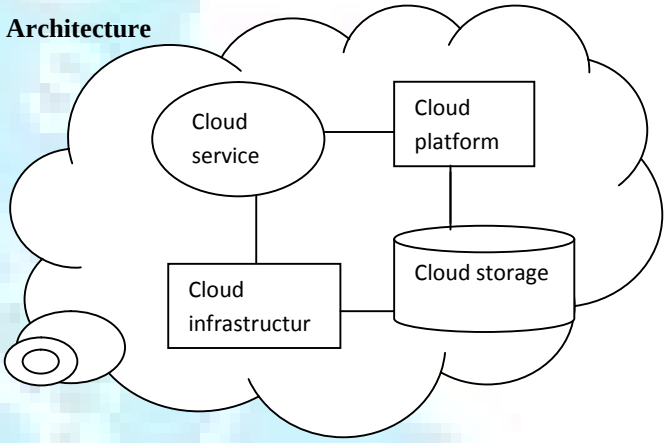


Fig 1: Cloud computing sample architecture

III. GAINING SECURE, SCALABLE DATA CLOUD COMPUTING

A. System Models

In this paper, we introduce new techniques to implement fine grained access control. In our techniques, the data is stored on the server in an encrypted form while different users are still allowed to decrypt different pieces of data as per the policy. This effectively preventing unauthorized data access. Similar to Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing [17], we assume that the system is composed of Cloud Servers, data owner ,data Consumers and a Third Party Auditor if necessary.it is basically a combination of many data owners and many data consumer . In order to access data files shared by the data owner, Consumers can download data files of their relevance from Cloud Servers and then decrypt. Neither the data owner nor users will always be online. They come online when the need arises. For simplicity, we assume that the sole access privilege for users is data file reading. Stretching our proposed scheme to support data file writing is trivial by asking the data writer to sign the new data file on each update as [12] does. Cloud Servers remains online and it is operated by the Cloud Service Provider (CSP). They are assumed to have heavy computation power and storage capacity . The Third Party Auditor is used for auditing every file access event remains online as well. we

also assume that the data owner can run his own code on Cloud Servers and store , manage his data files.

B. Security Models

In this work, we just consider Honest but Curious Cloud Servers as Over encryption: Management of access control evolution on outsourced data [14] does. We assume Cloud Servers are interested in access privileges information and contents of user files. For harvesting file contents Cloud Servers connive with a small number of malicious users when it is beneficial. it is assumed that Communication channel between the data owner/users and Cloud Servers secured under SSL. To achieve the goal of accessing the files with in and outside the access privileges, unauthorized users may work independently or cooperatively.

C. Design Goals

Our main design goal is to help the data owner achieve scalable, fine access control on files stored by Cloud Servers. Specifically, we enable the data owner to enforce a unique access structure ,that is designed over set of data files for each user which allows user to access some particular files. Our model also prevent data content and access privileges of each user of file being disclosed by Cloud server. By this design our proposed scheme also able to achieve user accountability ,revocation of user as a general one to many communication system requires. Goal of system scalability is being achieved by sense of efficient designing of our proposed scheme.

D. System design

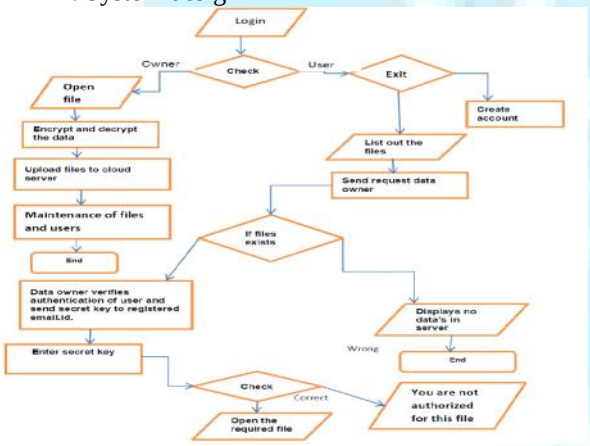


Fig 2: System Data Flow diagram

IV. TECHNIQUE PRELIMINARIES

In advanced attribute based encryption, attribute are associated with data files attributes for each of which a public key component is defined. This encryption is a public key cryptography primitive if we having communication one-to-many communication. Message by encrypting from its corresponding public key to key component having encrypt or in which the set of attributes .The interior node of access tree having threshold gates ANDs, ORs and Leaf node are associated with attributes and Each user is assigned an access structure which is usually defined as an access tree over data attributes. Each secret key is defined to reflect the access structure so that the user is able to decrypt a cipher text if and only if the data attributes satisfy his access structure [23]. A scheme is composed of four algorithms followed by atomic proxy re-encryption which can be defined as follows:

- Setup Attributes
- Encryption
- Secret key generation
- Decryption

A. Setup Attributes

This is a randomized algorithm, used to set attributes for users that takes no input other than the implicit security parameter. A bilinear group G1 of prime order p with a generator g, a bilinear map b : G1 × G1 → G2 . bilinear groups having properties of some property that are bilinear , computability, and not in the process of degeneracy. Following attributes are denoted as

U = {1, 2. . . N} Attributes of users
PK = (X, V1, V2, . . . , VN) Public key
MK = (x, v1, v2, . . . ,vN) Master key

where Vi ∈ G1 and vi ∈Zp are for attribute i, 1 ≤ i ≤ N, and X ∈ G2 is another public key component. We have Vi = gvi and X = b (g, g) x, x∈Zp. While PK is publicly known to all the parties in the system, MK is master key that is kept as a secret.

B. Encryption:

This is a algorithm follows randomization techniques,takes a set of attributes I as input ,a message and public key the cipher text E is output that is as follows:

E = (I, Ẽ, {Ei∈I })

where Ẽ= MXh, Ei = Si h. and h is randomly chosen from Zp

C. Decryption:

This algorithm takes as input the cipher text E which is encrypted under the attribute set I, the user's secret key for access tree V is SK , and the public key PK. It first computes b(Ei, ski) = b(g, g)pi(0)h. Then, using the polynomial interpolation technique it aggregates the results. Finally, the output the message M if and only if I satisfies V.

D. Access tree V:

Let V be a tree representing an access structure. As defined in access structure non-leaf node of the tree can be represented by threshold gate and a threshold value.. When kx equals to unit , the threshold gate is an OR gate and when kx = numx(Number of child) it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value kx = 1. To facilitate working with the access trees, we define a few functions. Parent(x) denotes parent of node X. The function att(x) is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in the tree.

E. Satisfying an access tree:

Let V be an access tree where root denotes by r. Vx the sub tree of V rooted at the node x. Hence V is the same as Vr. If a set of attributes I satisfies the access tree Vx, we denote it as Vx(I) = 1. We compute Vx(I) recursively as follows. If x is a non-leaf node, evaluate Vx'(I) for all children x' of node x. Vx(I) returns 1 if and only if at least kx children return 1. If x is a leaf node, then Vx(I) returns 1 if and only if att(x) ∈ I.

F. Construction of Access Trees:

In the process of construction of an access-tree V, cipher texts are labeled as set of attributes that define the descriptive attributes. In access tree each interior node of the tree is a threshold gate ANDs ,ORs and Only leaves are associated with attributes. If there is an assignment of attributes from the cipher texts to nodes of the tree such that the tree is satisfied ,user can decrypt a cipher text with a given key .

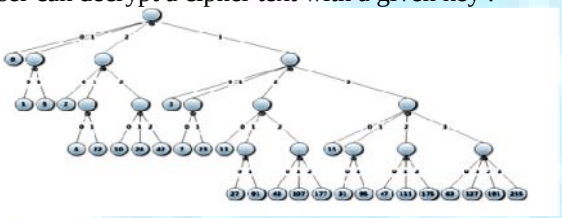


Fig 3: Access Tree Structure

G. Access Tree / Key-policy(T)

Access Policy to be associated with private key where leaf nodes are attributes coming from fuzzy identity. Cipher text has set of attributes, Keys reflect a tree access structure, Decrypt if attributes from CT satisfy key's policy.in current scenario Monotonic Access Formulas Tree of ANDs, ORs, threshold (k of N) and Attributes at leaves.

Fig 4: Access tree structure with threshold And, ORs and attributes at leaves.

Example:

Assuming, Alice has the following key policy

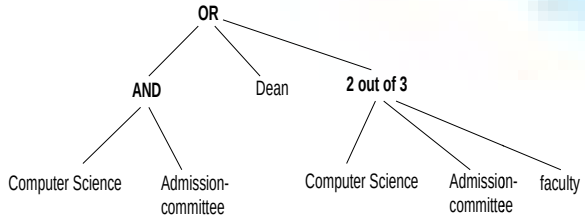


Fig 5: Access Tree Structure Example

Alice can decrypt a file encrypted with the attribute set {"Computer Science", "Admission committee"}. But she cannot decrypt another ciphertext associated with attributes {"Computer Science", "program-committee"}.

V. OUR PROPOSED SCHEME

A. Main Idea

Access control is a classic security topic which has a lot of study and proposed models. Unfortunately, these models are not applicable to cloud as the data owners and service providers are not in the same trusted domain. Therefore, it is required to proposed a new access control considering this issue to achieve scalable, flexible, fine-grained access control. Our proposed scheme enables the data owner to delegate tasks of data file creation ,encryption ,decryption ,re-encryption and user secret key update to cloud servers without disclosing data contents or user unique access structure information. We identify security challenges that arise in incorporation of cloud-based services, and present a set of solutions to address them. Main issues such as privacy, scalability for key management, flexibility in access and user revocation which are the most important considerations for gaining scalability and flexibility. We achieve our design goals by a novel structuring ,previously defined cryptographic primitives , Advanced attribute based encryption in which a unique access structure is assign for each attributes only this access structure decide the decryption key for file to be viewed. we can assume that we assign privileges to view a file. After completion of encryption process atomic proxy cryptography technique of re-encryption is used for revocation of user to update the attributes of all the live users. If we use encryption algorithm alone this would introduced have burden and heavy computational overhead such an issue caused by operation of user revocation , which require data owner to stay online and requires data owner to re encrypt all the accessible files to leaving user. In existing RLS, revoked user details such as private key are updated manually after each user revocation. Revoked users can access the cloud, But for efficient user revocation for dynamic groups Delta revocation list sets a ttp value (threshold value) , when it reaches the threshold value revoked users are updated automatically. Revoked users can't able to access the cloud hacking attack is reduced and communication overhead is also reduced, enable the data owner for computation operations to cloud Servers without disclosing the underlying file contents. This construction allows each data owner to access his data files with minimum online time and minimum overhead which the aim of our work. We formally prove the security of AABE based on security scheme by Bethencourt et al.[23] and analyze its performance and computational complexity. In atomic proxy cryptography two parties publish a proxy key that allows an untrusted intermediary to convert cipher texts encrypted for the first party directly into cipher texts that can be decrypted by the second the intermediary learns neither clear text nor secret keys by cloud server by this we can reduce key management ,key distribution load of data owner without disclose data content and user access structure .Main contributions of this paper can be summarized as follows.

1) To the best of our knowledge, In this paper we achieve scalability, data confidentiality, and Integrity simultaneously for data access control in cloud computing;
2) Our scheme enables the owner of data to get deal with most of tasks of computation to cloud servers without disclosing contents of data and user access structure information.

Definition and Notation

PK, MK	System public key and master key
Ti	Public key component for attribute i
ti	Master key component for attribute i
SK	User secret key
ski	User secret key component for attribute i
Ei	Cipher-text component for attribute i
I	Attribute set assigned to a data file
DEK	Symmetric data encryption key of a data file
P	User access structure
LP	Set of attributes attached to leaf nodes of P
AttD	The dummy attribute
UL	The system user list
AHLi	Attribute history list for attribute i
rki↔i'	Proxy re-encryption key for attribute i from its current version to the updated Version i'
δO,X	The data owner's signature on message X

Table.1 gives the description of notation to be used in our scheme.

VI. ANALYSIS OF OUR PROPOSED SCHEME

1. Security Analysis

We first analyze properties related to security of our proposed scheme that is as follows.

A. User Access Privilege Confidentiality : As per our scheme information of user regarding user access tree to disclosed to the leaf node only. Interior nodes of an access tree are unknown to cloud server and it can be any threshold gates. it is hard for Cloud Servers to recover the access structure and thus derive user access privilege information.

B. User Access Privilege Confidentiality: In our scheme consumer and data owner has to register himself then verification of each consumer is been done by data owner from cloud sever and each consumer have to register their valid Email-id.So whenever the data owner uploads one data, before that, the data owner takes key, public key and access tree as input and encrypts the data and uploads to server. Then the server verifies the user authentication details and store and also pass it to proxy server. Whenever the data consumer request for one data, the server verifies the user and sent the data to users registered mail-id. Then the data consumer takes cipher text, attributes and secret key as input and decrypts the data.

C. Fine-grainedness of Access Control : Flexible access structure for each user is define by Data owner which is able to define and enforce expressive. Specifically, the access structure of each user is defined as a logic formula over data file attributes, and is able to represent any desired data file set.

D. User Secret Key Accountability: This property can be immediately achieved by using the enhanced construction of Advanced-ABE which can be used to disclose the identities of key abusers.

2. Performance Analysis:

This section numerically evaluates the performance of our proposed scheme in terms of the computation overhead introduced by each operation as well as the cipher text size.

1)Computation Complexity: We analyze the computation complexity for the following operations:

a. System Setup: In this operation, bilinear group has been defined by the data owner and generate PK and MK .As is described, the main computation overhead for the generation of PK and MK is introduced by the N group multiplication operations on G1.

b. New File Creation : The main computation overhead of this operation is the encryption of the data file using the symmetric DEK as well as the encryption of the DEK using Advanced Attribute based encryption . The complexity of the former depends on the size of the underlying data file. The computation overhead consists of | I | multiplication operations on G1 and 1 multiplication operation on G2, where I denotes the attribute set I of the data file.

c. File Deletion : This operation involves the data owner as well as Cloud Servers. The former works to compute one signature at a time and the latter verifies this. For the execution of Granting a new user data owner, Cloud server and user is responsible .composition of the generation of the user secret key and encryption of the user secret key using the user's public key responsible for computation overhead . The former accounts for | L | multiplication operations on G1 , where L denotes the set of leaf nodes of the access tree. The main overhead for Cloud Servers is one signature verification. The user needs to do two PKC operations, one for data decryption and the other for signature verification.

d.User Revocation: User revocation is performed to satisfy access structure of Living user's for that data owner finds set of attributes that must to a minimal set. Next, the updating process of attribute redefining system masters key components in MK. first stage from two stages of user revocation is in between the owner of the data and Cloud Servers. The complexity of

algorithm AMinimalSet which is responsible for overhead is actually mainly contributed by the CNF conversion operation . we are Assuming that D is size of the minimal set that is returned by AMinimalSet , the computation overhead for execution of algorithm AUpdateAtt is D≤N, contributed by D number of multiplication operations on G1.In addition, to compute D signatures on public key components ,the data owner needs to compute it using size of D.by this overall computation overhead at this stage on Cloud Servers is negligible.The complexity of user revocation is counting , Use N instead of choosing the size of the access structure we use N ,since in practical scenarios algorithm of AMinimalSet is efficient if we limit the size of access structure and it didn't affects system scalability, but on G1 cost of each signature or multiplication operation is expensive.

e. File Access : This operation occurs between Cloud Servers & user. . In the worst case, the algorithm AUpdateSK would be called | L | − 1 times, which represents | L |− 1 multiplication operations on G1 . Each execution of the algorithm AUpdateAtt File accounts for one multiplication operation on G1 . In the worst case, Cloud Servers need to call AUpdateAtt File N times per file access. Our re-encryption technique and threshold value solution will greatly reduce the average system- wide call times of these algorithms from statistical point of view. File decryption needs | L | bilinear pairing in the worst case.

Cipher text Size: The cipher text consists of an unique ID number , a header name, and a body. The body is block of data. The header for each data file is composition of I attribute set , one group element on G2 and | I | group elements on G1.

We measure performance of our proposed scheme over following parameters/metrics:

1.Computation overhead at data owner– In current scenario of mobile cloud computing and public cloud storage, the emphasis is on low computational need on data owner's side. In our scheme we introduce Advanced attribute based encryption in which data owner assign a unique access structure by this key structure secret is for decrytion is designed, so that only authorized user can access the requested file. With in this we have removed the expiration time from revocation concept, instead of this we defined a TTP value by this overhead of key distribution , key management and re encryption of keys reduced .

2.Computation overhead at CSP – The cloud service provider is thought to be resourceful in terms of computation power. Yet the burden of processing imposed by security measures should not be very high. So, for any technology to be well-accepted, it should be computationally light. In our architecture at server side a ttp value (threshold value) is set, when it reaches the threshold value revoked users are re-encryption is performed revocation can only be done when. Revoked users can't able to access the cloud hacking attack is reduced and communication overhead is also reduced. is connected to one proxy server and the server contains access policy to allow user to access by this all the overhead is of key management is passes to proxy server.

VII. RESULTS

Summarizes the Operational computation complexity of our proposed scheme.

File Creation	O (| I |)
File Deletion	O (1)
User Grant	O (| L |)
User Revocation	O (N)
File Access	O (max (| L | , N))

Fig 6. : Complexity of our proposed scheme

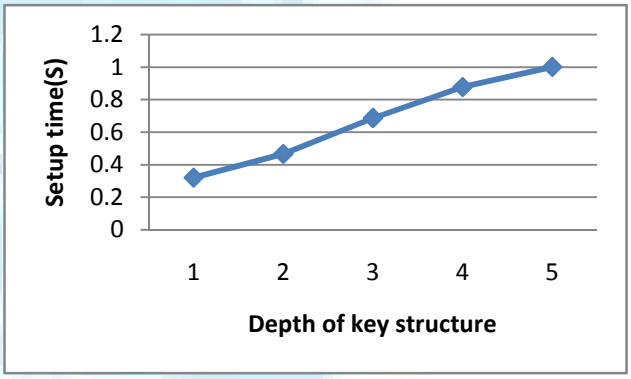


Fig. 7 shows the time required to setup the system for a different depth of key structure

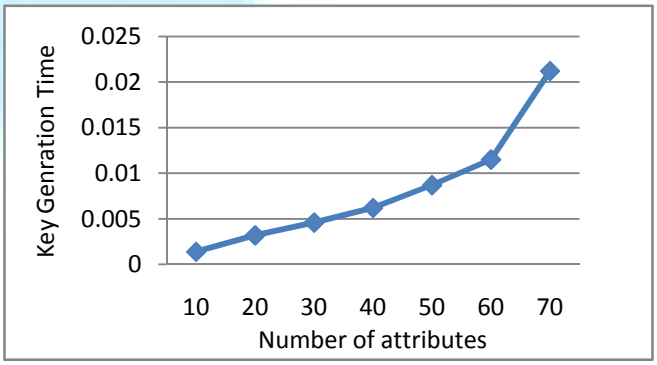


Fig8 Shows time of key generation with respect to number of attributes at varying level of threads

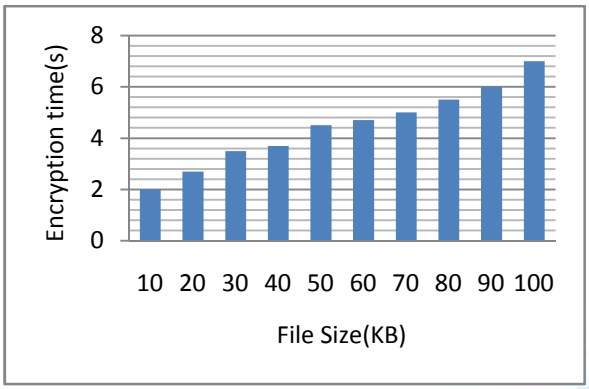


Fig 9 Time Taken for Encryption of File

Access Tree Level	Decryption time(s) (Existing)	Decryption time(s) (Our scheme)
1	0.65	0.428
2	0.64	0.446
3	0.63	0.463
4	0.64	0.484
5	0.65	0.502

Table 2 Experiments on file decryption/Creation

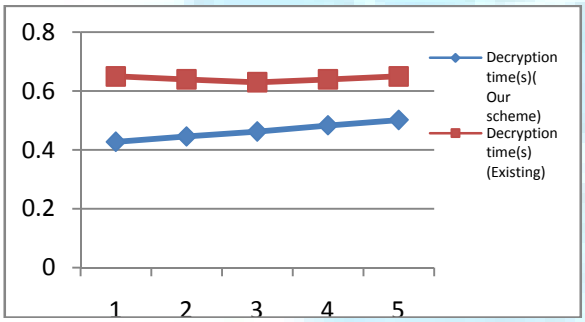


Fig 10 Time Taken for decryption /creation of File

This scheme setup in Java and we have shown in diagram the comparison with increasing number of request k between the overall system performance of existing scheme with our proposed scheme.

k	Existing system	Our system
1	0.47	0.438
2	1.76	0.632
3	2.95	0.784
4	3.89	1.53
5	5.12	2.675
6	7.26	3.725

Table 3: Comparison between existing scheme and our proposed scheme

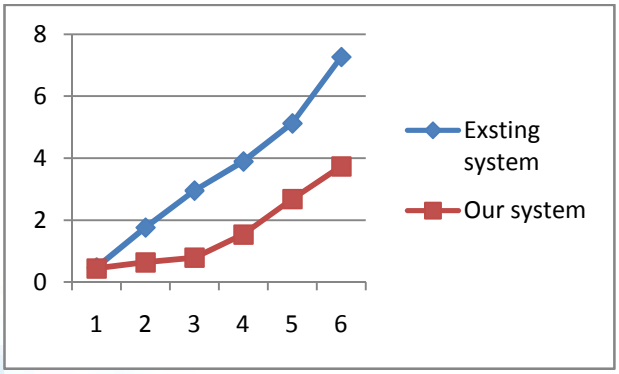


Fig 11:Comparison between existing scheme and our proposed scheme

VIII. CONCLUSION

This paper aims at fine-grained data access control in cloud computing. One challenge in this context is to achieve fine- grainedness, data confidentiality and scalability simultaneously, which is not provided by current work. In this paper we propose a scheme to achieve this goal by exploiting Advanced- ABE, in which we assign a threshold value for revocation of user, as the no of user reaches threshold value all the users key components updated except revoked. For updation process we perform atomic proxy re encryption technique . Moreover, our proposed scheme can enable the data owner to deal with computation overhead to powerful cloud servers. Confidentiality of user access privilege. Formal security proofs show that our proposed scheme is secure under standard cryptographic models.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” University of California, Berkeley, Tech. Rep. USB-EECS-2009-28, Feb 2009.

[2] Amazon Web Services (AWS), Online at http://aws. amazon.com.

[3]Google App Engine, Onlineat http://code.google.com/appengine/.

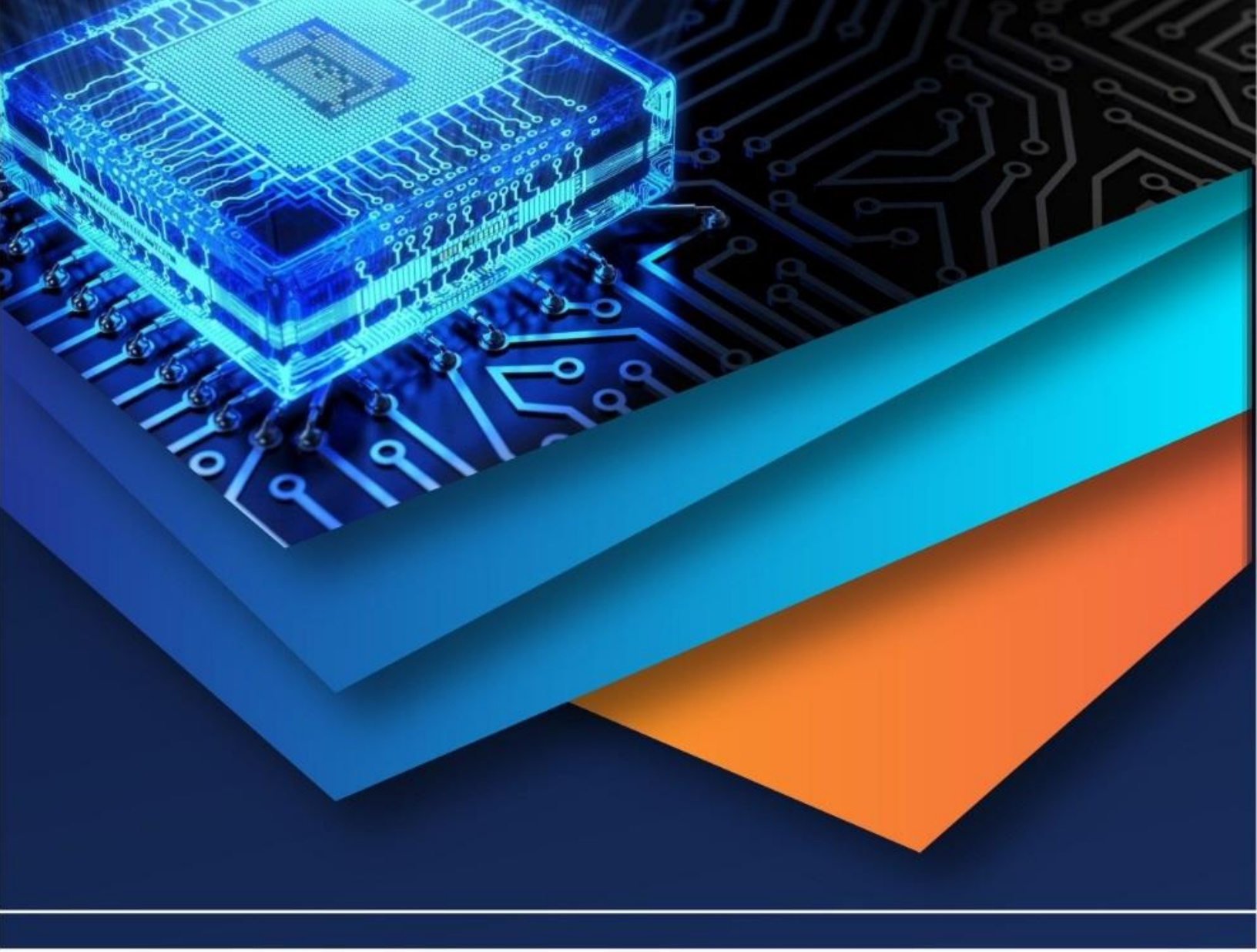
[4] Microsoft Azure,http://www.microsoft.com/azure

[5] 104th United States Congress, “Health Insurance Portability and Accountability Act of 1996 (HIPPA),” Online at http://aspe.hhs.gov/ admnsimp/pl104191.htm, 1996.

[6] H. Harney, A. Colgrove, and P. D. McDaniel, “Principles of policy in secure groups,” in Proc. of NDSS’01, 2001.

[7] P. D. McDaniel and A. Prakash, “Methods and limitations of security policy reconciliation,” in Proc. of SP’02, 2002.

[8] T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in Proc. of SP'03, 2003.
[9] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. of CCS'05, 2005.
[10] J. Anderson, "Computer Security Technology Planning Study," Air Force Electronic Systems Division, Report ESD-TR-73-51, 1972, http: //seclab.cs.ucdavis.edu/projects/history/.
[11] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Scalable secure file sharing on untrusted storage," in Proc. of FAST'03, 2003.
[12] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in Proc. of NDSS'03, 2003.
[13] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy reencryption schemes with applications to secure distributed storage," in Proc. of NDSS'05, 2005.
[14] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in Proc. of VLDB'07, 2007.
[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. of CCS'06, 2006.
[16] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in Proc. of EUROCRYPT '98, 1998.
[17] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS '09, 2009.
[18] L. Youseff, M. Butrico, and D. D. Silva, "Toward a unified ontology of cloud computing," in Proc. of GCE'08, 2008.
[19] S. Yu, K. Ren, W. Lou, and J. Li, "Defending against key abuse attacks in kp-abe enabled broadcast systems," in Proc. of SECURECOMM'09, 2009.
[20] D. Sheridan, "The optimality of a fast CNF conversion and its use with SAT," in Proc. of SAT'04, 2004.
[21] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in Proc. of CRYPTO'01, 2001.
[22] M. Atallah, K. Frikken, and M. Blanton, "Dynamic and efficient key management for access hierarchies," in Proc. of CCS'05, 2005.
[23]J. Bethencourt, A. Sahai, and B. Waters, "Cipher text-policy attribute based encryption," in Proc. IEEE Symp. Security and Privacy, Oakland,
CA, 2007.
[24]Zhiguo Wan, Jun'e Liu, and Robert H. Deng, "HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing", IEEE Transactions on Information Forensics and Security, Vol. 7, No. 2, April 2012.
[25] Vibha Sahu, Brajesh Dubey, Dr.S.M. Ghosh " Clouding Computing Threats", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2 Issue VIII, August 2014, Page No: 207-213
[26] K. Vijesh, P. Santhadevi "Cloud Computing: A Beginners Primer", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2 Issue VII, July 2014, Page No: 42-52
[27] Sonam Sudha, Ms.Vasudha Arora "Identity and Access Management in Cloud Computing", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2 Issue VII, July 2014, Page No: 146-153
[28] Anil Behal, Dr. Harish Rohil "Data Encryption Using Cloud Computing", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2 Issue VII, July 2014, Page No: 234-241
[29] B. Pavan Kumar, Prof. GVNKV Subba Rao "Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud", International Journal for Research in Applied Science and Engineering Technology (IJRASET), Volume 2 Issue VII, July 2014, Page No: 273-278



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY