DATA INTEGRITY PRESERVATION IN CLOUD COMPUTING BASED ON SERBAC

Sunitha B S[1] and Anirban Basu[2]
[1]Department of Information Science & Engineering, EPCET, VTU, Research Scholar
[2]Department of Computer & Engineering, Professor, APSCE, Bangalore, India

ABSTRACT
Cloud computing and services are rapidly developing, there has been a growing trend to use the cloud for large-scale data storage. The usage of cloud services has raised the important security issue of how to control and prevent the data stored in the cloud. However, various existing approaches have a major shortcoming, as they assume the server is trustworthy and require complete disclosure of sensitive location information by the user. In this work we describe a method for fully distributed authentication using public cloud based on the Session Authentication ticket framework. Specifically, in our scheme, the authentication takes place for every user with a role along with a request. The session authenticator service provider verifies the tokens and evaluates the roles based on dual encryption mechanism for the user thus providing the efficiency. In this paper, we define the protocols based on the data integrity preservation and the dual encryption mechanism based on our efficient access control system $SERBAC$.

KEYWORDS: *Access control Roles, Broad group key Management, RuleSet*

I. INTRODUCTION

This Cloud computing is one of the most valuable innovations for business, providing cheap, virtual services that once required expensive and local hardware. It also provides services for users to build deploy and manage their applications on the cloud. It involves virtualization of resources that maintains and manages by itself. We place almost everything in the cloud, but what do we really know about its security? How do we protect ourselves and our privacy from being compromised? Security is mainly necessary for strong privacy in all online computing factors, but security alone is not enough. Security and cost are the top issues in this field and they vary greatly, depending on the vendor one choose. Despite the first success and recognition of the cloud computing model and the extensive availability of providers and tools, a number of challenges and risks are innate to this new model of security service in cloud computing. An approach to mitigate these concerns is the use of encryption. Even though, encryption assures the confidentiality of the data against the cloud, the use of conventional encryption approaches is not sufficient to support the enforcement of fine-grained organizational access control roles (ACRs). Our framework is based on improving collaboration between cloud providers, service providers and service consumers in managing the security of the cloud platform and the hosted services. Thus, the session authentication is proposed. In this Session Authentication service, a dual encryption mechanism is followed which is a well-established authenticated system. To demonstrate this framework, two encryption methods have been proposed. So for each user in the $SERBAC$, the encryption takes place from the session authentication server which is more efficient compared to the other encryption mechanisms. The research paper is organized as follows. Section two discusses the related work. The initialization and tunneling mechanism of security and the data integrity is presented in section three. Section four discusses the security of roles and dual encryption mechanism security mechanism in $SERBAC$. The experimental results are presented in section four. The concluding remarks are discussed in the last section of the paper.

II. RELATED WORK

Much work has been done in the Data Integrity preservation in Cloud Computing sector. Let us look into some of the survey which exists. In the literature, there exist many hierarchy access control schemes [22] [3] which have been constructed based on hierarchical key management schemes, and approaches using hierarchical key management schemes to enforce access control of roles for data storage are discussed in [7] [10]. However, these solutions, also [8] have several limitations. For instance, if there is a large number of data owners and users involved, the overhead involved in setting up the key infrastructure can be very high indeed. Furthermore, when a user's access permission is revoked, all the keys known to this user as well as all the public values related to these keys needs to be changed, which makes these schemes impractical. An alternative approach for the management of keys is Hierarchical ID-based Encryption, such as [12], [13]. However, in a Hierarchical ID-based Encryption scheme, the length of the identity becomes longer with the growth in the depth of hierarchy. In another survey, first attribute-based encryption scheme was proposed in [11] based on the work in [14], and some other attribute-based encryption schemes have been proposed afterwards. In these schemes, data is encrypted to a set of attributes, and users who have the private keys associated with these attributes can decrypt the data. These works have provided an alternative approach to secure the data stored in a distributed environment using a different access control mechanism, such as [9]. In [15], it is shown that an attribute-based encryption scheme can be used to enforce access control of roles. However, in that approach, the size of user key is not constant, and the revocation of a user will result in a key update of all the other users of the same role. [16] Also investigated the solutions of using attribute-based encryption scheme in access control model. However their solution only maps the attributes to the role level in controlling the access [1], and they assumed that the access control system itself would determine the user membership. Even though these attribute-based constructs are expressive and provably secure, they are not suitable for group management and especially in supporting forward security when a user leaves the group and in providing backward security when a new user joins the group. Other approaches to protect data privacy in a cloud environment include using direct encryption and proxy re-encryption. In these cryptographic schemes, data is allowed to be encrypted directly to the users with whom the owners wish to share the data [17], [18]. This is analogous to the access control roles in Discretionary Access Control (DAC) model. Hence they are usually used in systems where DAC model is adopted. Since the permissions in such systems are specified either in a flat out structure or in an access matrix, we do not compare them with our schemes as the access roles are specified differently in SERBAC model. Recent research efforts, [21] [4] have proposed approaches to construct privacy preserving access control systems using a third-party storage service. In such approaches, the data owner has to enforce the access control of roles and the privacy of the users from the content publisher is not protected. Further, in some approaches [5][6], multiple encryptions of the same document are required which is inefficient. Recently Liang et al. [26] has extended the traditional PRE to attribute based systems and independently Chu et al. [27] has extended the traditional PRE to support conditions where a proxy can re-encrypt only if the condition specified by a public key on a third party is satisfied. However, they do not protect the identity attributes of the users who access the system [2] and are difficult to manage. Lan Zhou et al [23] present a design of a trust-based cloud storage system, which shows how the trust models can be integrated into a system that uses cryptographic RBAC schemes. Wenhui Wang et al [24] paper suggests an adaptive access algorithm by introducing the trust into cloud computing to decide the access control to the resources using an improved RBAC technique to solve more complex and difficult problems in the cloud computing environment. Jiangfeng Li et.al[25] presented 4D-role based multitenancy model is proposed for running various applications and services in the multitenancy cloud platform To overcome these issues, we have proposed a Data Integrity preservation system secure and accessible in the public cloud. A high secure performance access control role evaluation mechanism is proposed based on $SERBAC$.

III. BUILDING BLOCKS

Here, Security Initialization schemes are discussed as well as and the Secure tunneling mechanism. Abstract view of the main algorithms of those protocols and how it is used to build the privacy-

preserving attribute based group key management in data integrity preservation scheme are briefed. And then, an overview of securing the roles and the dual encryption security mechanism approach in $SERBAC$ based on the Session service is given.

3.1 Security Initializations

It solves the problem of how efficiently to encrypt a message and broadcast it to a subset of the cloud users in a system. The subset of users can change dynamically. The users are called privileged, in the broadcast encryption and the non-authorized users called revoked. We denote the set of cloud users by C, the set of revoked users V. The set of privileged users is thus, C / V. We set $A = |C|$ and $v = |V|$. The broadcast encryption follows the process where the message for each privileged user is encrypted separately and then broadcasting all the encrypted messages which is very inefficient since the message length is very large which is given by $(O(A - v))$.Thus subset-cover algorithm that supports broadcast encryption with stateless users is used. The algorithm builds a binary tree and assigns users to the leaf nodes and thus results in a predefined user grouping. Each such group is called a subset. A user can be a member of several subsets. The cover, denoted by R, is defined as the set of subsets that contains all the privileged users, that is, users in C / V. The subsets in the cover are disjoint and hence each privileged user belongs to only one subset. A subset-cover based broadcast encryption is established based on the following algorithms: Setup, GetSecKeys, GetCover, Broadcast, KeyDer and Decrypt. Each of the algorithms are defined as follows.

Setup(b, L): The server constructs a binary tree T wherethere are at least L leaf nodes for the security parameterb which denotes the bit length. Each node in T is eitherassigned a unique key whose length is decided by b,or can computationally derive a unique key. The user c_i, $i = 1,2, \dots, L$, is assigned the i^{th} leaf node.

GetSecKeys (c_i): The server gives all the key assigned to c_i in T.

GetCover (C / V): Given the privileged user setC / V, the server outputs the cover R, that is, the set of disjoint subsets that cover all the privileged users.

Broadcast (M, R): The server generates a session key Y andencrypts the message G with Y and encrypts Y with each keyin the cover R.

KeyDer (c_i , R): The user cidentifies its subset in the cover R, outputs the key that decrypts the session key.

Decrypt (R, Y): It decrypts the encrypted message M with the key Y, to output the message G.

Here, we consider the complete sub tree algorithm. The complete sub tree algorithm improves the basic technique for simultaneously revoking v users and describingthe privileged users usingv $\log(A/v)$ subsets.

3.2. Secure tunneling mechanism

It obliviously delivers a message to the users who satisfy certain conditions. This protocol consists of three entities: a server S, a cloud user C and a trusted third party called the identity provider P. The Tunneling Mechanism is established based on the algorithms: Setup, GenCom and GetData. These algorithms are described below.

Setup (b): The P runs a Pedersen commitment setup protocol to generate the system parameters, a finite cyclic group $\mathbb{G}$of large prime order m, two generators x and y of $\mathbb{G}$. The size ofm is dependent on the security parameterb.

GenCom(a): A C wants to commit to the value a. It submitsa to the P. The P computes the Pedersen commitment$t = x^a y^v$.wherev is randomly chosen from $\mathbb{F}_m$. The P digitallysigns t and sends v, t and the signature of t to the C.

GetData$(t, cond, r)$: The C sends the signed commitmentt and indicates the S's condition cond that it wants to satisfy. cond has the format "name predicate value" where the predicate can be $\geq, >, \leq, <$ or $=$. After an interactive session,the S encrypts the data r and sends the encrypted data,called envelope, to the C. The C can decrypt and access thedata only if it satisfies the condition.

The following properties are carried by the OCBE protocols. The S does not learn the identity attributes of the users. A C can open the envelope only if it's committed attribute value satisfies the

condition. A C cannot submit fake commitments in order to satisfy a condition as the commitments are signed by the P.

3.3. Data Integrity Preservation

Here we use the Broadcast Group Key Management and Privacy Preserving Attribute Based-Group Key Management protocols. The overall construction is based on the Attribute Based-Group Key Management scheme which is an expressive construct of the access control vector Broadcast Group Key Management scheme. The Broadcast Group Key Management extends the Group Key Management where the rekey operation is performed with a single broadcast without requiring the use of private communication channels. TheBroadcast Group Key Managementschemes do not give users the private keys, instead users are given a secret which is combined with public information to obtain the actual private keys. Such schemes have the advantage of requiring a private communication only once for the initial secret sharing. The subsequent rekeying operations are performed using one broadcast message. Further, in such schemes achieving forward and backward security requires only to change the public information and does not affect the secret shares given to existing users. The establishment of Broadcast Group Key Management scheme consists of the following five algorithms : Setup, SecGen, KeyGen, KeyDer, and ReKey. The establishment of Attribute Condition is as follows. An attribute condition Ais an expression of the form: $\mathrm{iden}_{\mathrm{attr}}$ qtwhere nameisthe name of an identity attribute attr, q is a comparison operatorsuch as $=, >, \geq, <, \leq, \neq$ and t is a value that can beassumed by the attribute attr. The establishment of Access Control Roles is as follows. An access control roles ACR is a set (e, d). Where, d denotes a set of data items$\{D_1, \ldots, D_t\}$and e is a monotonic expression over a set of attribute conditions that must be satisfied by a Cto have access to d. The ACR is embedded in an access structure A . Ais a tree in which the internal nodes representthreshold gates and the leaves represent Broadcast Group Key Management instances for the attributes. The goal of the access tree is to allow the derivation of the group key for only the users whose attributes satisfy the access structure A.Each threshold gate in the tree is described by its child nodes and thresh hold value. The threshold valuetof a node a specifies the number of child nodes that should be satisfied in order to satisfy the node. The root of the tree contains the group key and all the intermediate values are derived in a top down fashion. A user who satisfies the access tree derives the group key in a bottom-up fashion. Due to space constraints, the abstract algorithms of the Privacy Preserving Attribute Based-Group Key Management are provided. Thus, the Privacy Preserving Attribute Based-Group Key Management is established with the algorithms: Setup, SecGen, KeyGen, KeyDer and ReKey.

Setup (p, S, S_a): It takes the security parameter p, the maximumgroup size S, and the number of attribute conditions $S_a$as input, initializes the system.

SecGen(β): The secret generation algorithm gives a $C_i, 1 \leq i \leq S$ a set of secrets for each commitment $com_a \in \beta,, 1 \leq i \leq m$. It invokes Security Initialization:: GetSecGen and Secure tunnelling Mechanism::GetData algorithms.

KeyGen(ACR): The key generation algorithm takes the access control rolesACRas the input and outputs a symmetric key Y, a set of public information set μ and anaccess tree A . It invokes Security Initialization:: GetCover() and Secure Tunnelling Mechanism:: KeyGen algorithms.

KeyDer(α, μ, A): Given the set of identity attributesα,the set of public information set μ and the access tree A, the key derivation algorithm outputs the symmetric Yonly if the identity attributes in α satisfy the access structureA . It invokes Security Initialization::KeyDer and Secure Tunnelling Mechanism::KeyDer algorithms.

ReKey(ACR): The rekey algorithm is similar to the KeyGen algorithm. It is executed whenever the dynamics in the system change.

IV. OVERVIEW

Here we mainly focus on the Dual Encryption mechanism. It consists of the four entities: Session Server Authenticator, User, Identity Protocol and Cloud. However, unlike the SLE approach, the Server and the cloud collectively enforce ACR's by performing two encryptions on each data item. This dual layer enforcement allows one to reduce the load on the authenticator and delegates as much

access control enforcement duties as possible to the cloud. Thus, providing a better way to handle data updates, and user dynamics changes.

4.1 Optimization of Roles

The Server authentication incurs a high communication and computation overhead since it has to manage all the authorizations when user dynamics change in the SLE approach. If the access control related encryption is somehow delegated to the cloud, the Authenticator can be freed from the responsibility of managing authorizations through re-encryption and the overall performance would thus improve. Since the cloud is not trusted for the confidentiality of the outsourced data, the Authenticator has to initially encrypt the data and upload the encrypted data to the cloud. Therefore, in order for the cloud to allow to enforce authorization roles through encryption and avoid re-encryption by the Authenticator, the data may have to be encrypted again to have two encryption layers. We call the two encryption layers as inner encryption layer (IEL) and outer encryption later (OEL). IEL assures the confidentiality of the data with respect to the cloud and is generated by the Authenticator. The OEL is for fine-grained authorization for controlling accesses to the data by the users and is generated by the cloud. The TLE approach manages how to distribute the encryptions between the Authenticator and the cloud. There are two possible extremes. The first approach is for the Authenticator to encrypt all data items using a single symmetric key and let the cloud perform the complete access control related encryption. The second approach is for the Authenticator and the cloud to perform the complete access control related encryption twice. The first approach has the least overhead for the Authenticator as the Authenticator does not manage any attributes and perform fine grained access control related encryption. However it has the highest information exposure risk due to collusions between cloud user C and the cloud as one malicious cloud user C revealing the Authenticator's encryption key exposes all sensitive data to the cloud. Further, IEL updates require re-encrypting all data items. The second approach has the least information exposure risk due to collusions as the fine grained access control is enforced in the first encryption. However, it has the highest overhead on the Server as the Authenticator has to perform the same task initially as in the one layer Encryption approach and, further, needs to manage all identity attributes. An alternative solution is based on decomposing ACR's so that the information exposure risk and key management overhead are balanced. The problem is then how to decompose the ACR's such that the *Session Server* has to manage the minimum number of attributes while delegating as much access control enforcement as possible to the cloud without allowing it to decrypt the data.

4.2. Role Secure

We define the role secure problem as the optimization problem of finding the minimum number of attribute conditions that "covers" all the ACR's in the ACRB. We say that a set of attribute conditions covers the ACRB if in order to satisfy any ACR in the ACRB, it is necessary that at least one of the attribute conditions in the set is satisfied. We call such a set of attribute conditions as the attribute condition cover.

4.3. Dual Encryption Security Mechanism in SERBAC

The system consists of the four entities, Session Server Authenticator, User, Identity Protocol and Cloud. Let the maximum number of users in the system be S, the current number of users ben($< S$), and the number of attribute conditions S_a. The six phases of the dual encryption security mechanism are described below.

4.3.1. Session Setup

P's are trusted third parties that issue Session token to cloud user C's based on their identity attributes. It should be noted that P's need not be online after they issue tokens. A Session token, denoted by IT has the format$\rho, \text{tag}, P, \varphi$, where ρ is a pseudonym uniquely identifying aC in the system, tag is the name of the attribute,P is the Pedersen commitment for the Kerberos attribute valuea and $\varphi$is the P's digital signature on $\rho, \text{tag}, \text{and } P$.

4.3.2. Role Decomposition

Using the role decomposition, the authenticator decomposes each ACR into two sub ACR's such that the Session server authenticator enforces the minimum number of attributes to assure confidentiality

of data from the cloud. The algorithm produces two sets of sub ACR's, $ACRB_{server}$ and $ACRB_{cloud}$.TheSession server authenticator enforces the confidentiality related sub ACR's in $ACRB_{server}$and the cloud enforces theremaining sub ACR's, $ACRB_{cloud}$.

4.3.3. Session Registration

The cloud user C register their Session token IT to obtain secrets in order to later decrypt the data they are allowed to access. The user C register their IT's related to the attribute conditions with the Session Server, and the rest of the tokens related to the attribute conditions with the cloud using the Data Integrity::SecGen algorithm. When user C register with the Session Server, the Server issues them two sets of secrets for the attribute conditions that are also present in the sub ACR's in ACRB. TheServer Authenticator keeps one set and gives the other set to the cloud. Two different sets are used in order to prevent the cloud from decrypting the authenticator encrypted cloud data.

4.3.4. Preserving the Data Integrity

The Session Server encrypts the data based on the sub ACR's in ACRB and uploads them along with the corresponding public information sets to the cloud. The cloud in turn encrypts the data again based on the sub ACR'sinACRBServer. Both the Server and the cloud execute Data Integrity::KeyGen algorithm individually to first generate the symmetric key, the public information set PI and access tree A for each sub ACR. The Server arranges the sub ACR's such that each data item has a unique ACR. Note that the same role maybe applicable to multiple data items. Assume that the set of data items $D = \{d_1, d_2 \dots, d_i\}$ and the set of subACR'sACRB$_{server} = \{ACR_1, ACR_2, \dots, ACR_i\}$.The Server assigns a unique symmetric key, called an$\mathbb{K}$ key,$Y_n^{\mathbb{K}}$ for each sub$ACR_n \in ACRB_{server}$, encrypts all related data with that key and executes the Data Integrity::KeyGen to generate the publicPIand A.The Server uploads those encrypted data$\left(id, \mathcal{E}_{Y_n^{\mathbb{K}}}(d_n), n\right)$ along with theindexed public information sets (n, PI_n, A_n), wheren $= (1,2, \dots, i)$, to the cloud. The cloud handles the key management and encryption based access control for the ACR'sin$ACRB_{cloud}$. The cloud user C download encrypted data from the cloud and decrypt twice to access the data. First, it removes the encryption layer added by the cloud and then by the Server thus deriving the Inner Layer Encryption key using the Data Integrity::KeyDer algorithm. These two keys allow cloud user C to decrypt a data item only ifthe user C satisfies the original ACR applied to the data item. User credentials may change over the time. Further, already encrypted data may go through frequent updates. When the ACR's is modified, the Session Server has to perform the role decomposition again. If the set ACRB remains unchanged after the role decomposition, the Server does not have to re-encrypt and only needs to provide the updated ACRB to the cloud which enforces the new set of ACR's by re-encryption. Otherwise, both the Server and the cloud are required to re-encrypt all the affected data items.

V. PERFORMANCE EVALUATION

The security mechanism, Data Integrity Preservation SERBAC model has been developed for highly competitive and secured cloud computing environment. The system model presented has been developed on Visual Studio 2010 framework 4.0 with C#. The overall system has been developed and implemented with Amazon S3 cloud platform. The developed system has been simulated for different performance parameters like role computation overhead, user creation computation and storage overhead for the cloud user based on the Session Authentication. The relative study for these all factors has been performed against the existing mechanisms. This system or model performance has been verified for various user size with dynamic role assignments and the relative throughput as well as performance parameters have been checked for its robustness justification. Here, the role decomposition takes place and the data integrity is preserved. The dual encryption mechanism is been processed for every user. We have created the system where the RuleSet can have the maximum number of 8. Based on this RuleSet, various roles and users can be created for the performing the task. Users are assigned certain set of rules. Then, the Session authentication service takes place for each and every user, preserving the data.

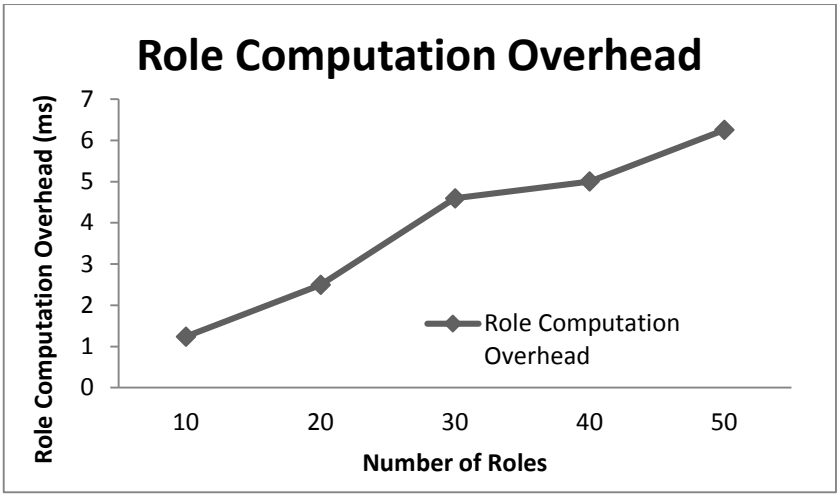

Figure 1. Role Computation Overhead

Based on the simulated data, the graph (Figure 1) is plotted for the role computation overhead in milliseconds based on the respective role assignments from 10 to 50 on the RoleSet based on the simulation workloads. Here, we consider maximum of 8 Roles. Thus, for every role creation, the decomposition takes place, thus, securing the roles which is optimized. The time taken is also predicted. When compared with uml based computation, this proves its efficiency

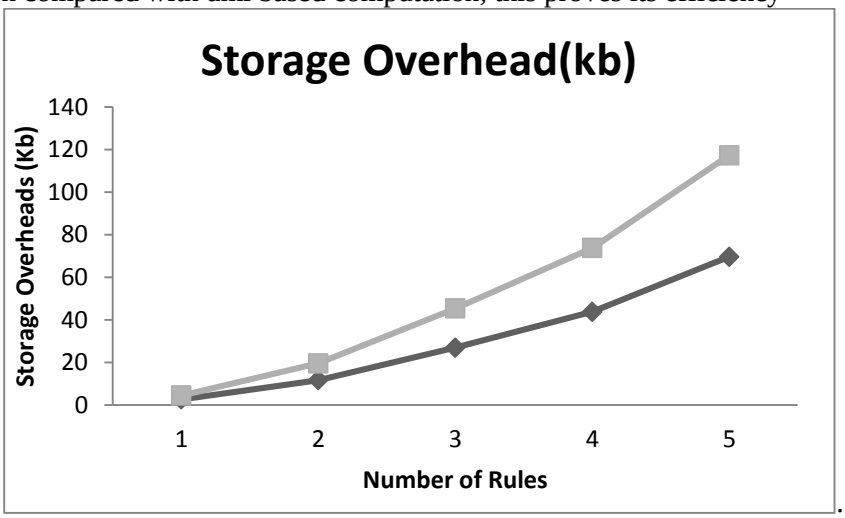

Figure 2. Storage Computation

This above graph (Figure 2) is plotted making the comparison of the storage overhead of our proposed system security in SERBAC against the existing system [9]. From the figure it is clear that the storage overhead is better even though the number of roles is increased. In the existing system, the xml sheets are created for each and every user and also for the permission of roles where it consumes lot of storage as well as the computation time. Whereas in our proposed system, the RoleSets for the multiple roles are already been computed based on the binary format of request and then it is assigned to the users which results in extreme minimization of storage space. Hence, the secureness is predicted based on the encryption which takes place twice. For each and every user, the Session Authentication takes place in which it verify the tokens and evaluates the roles. Let us consider a large user case scenario where the users are operated on Amazon S3 Cloud. Let the number of users be 100(n=100). The computation overhead results are obtained based on various parameters like role computation, user creation computation, storage computation and cumulative user overhead considering different graphs shown below.

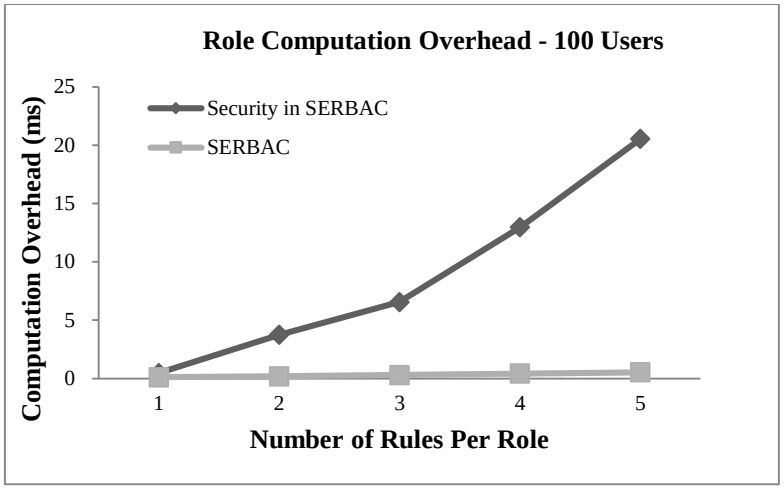

Figure 3. Role Computation Overhead Comparison

The Role Computation Overhead comparison for the 100 Users have been given in the graph above (Figure.3) based on the SERBAC and the data integrity preservation in SERBAC. As the role increases, the computation also increases, but it is seen to be very less in millisecond. Even though the roles are 25, the computation overhead is 0.35ms which is very less. But whereas in the encryption based mechanisms, huge computation takes place for the creation of roles which are in sec. Hence, proving our system to be more logical and productive.

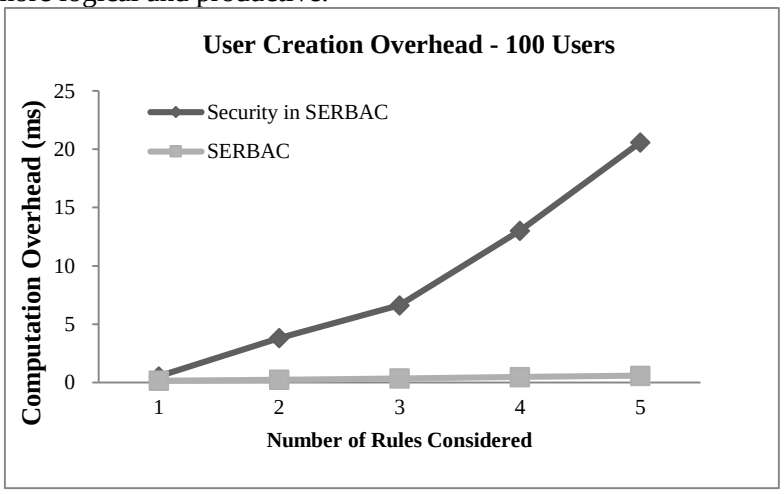

Figure 4. User Computation Overhead Comparisons

The graph (Figure.4), portrays the overhead computation of creating the users. In our system, the roles are optimized and then it is secured. The initialization of security based on the Session Authentication is been done. Thus, the users are created and then the Session Authentication takes place on the dual encryption mechanism preserving the complete data based on the Session Authentication Server. Here, it can be seen that the overhead is slightly greater compared to the user creation in SERBAC. This is because of the authentication done on the each user. Even then it can be considered because of its efficiency compared with other encryption mechanisms. Based on these results, we have verified that the proposed model can be an effective, robust optimum and very secure for role based access control in cloud environment.

VI. CONCLUSION

We enhanced the extendable role-based access control (RBAC) mechanism which plays a critical role in access control system in a large set of roles (rules) and users in the cloud. Our paper proposes a session authentication where the dual encryption mechanism takes place based on the SERBAC. Experimental results show that this method is most secure, feasible and efficient which consumes less

time and storage capacity even though permitting large number of users and roles and importantly more secure with the Session Authentication. Finally, we conduct comprehensive performance analysis, which shows that our scheme is more secure, efficient and practical than existing schemes

VII. FUTURE SCOPE

Further the work is carried towards provisioning security in $SERBAC$. The insights of security and access to Cloud Computing need to discuss. Despite the popularity of cloud services and their wide adoption by enterprises and governments, cloud providers still lack security services that guarantee both data and access control consistency across multiple data centers

REFERENCES

[1] Min Xu, DumindaWijesekera, Senior Member, IEEE, and Xinwen Zhang, Member, IEEE Runtime Administration of an RBAC Profile for XACML. Dec. 2011
[2]Mohamed Nabeel, Member, IEEE, Ning Shang, and Elisa Bertino, Fellow, IEEE Privacy Preserving Policy-Based Content Sharing in Public Clouds, Nov. 2013
[3]H. R. Hassen, A. Bouabdallah, H. Bettahar, and Y. Challal, "Key management for content access control in a hierarchy," Computer Network , vol. 51, no. 11, pp. 3197–3219, 2007.
[4]S. Coull, M. Green, and S. Hohenberger, "Controlling Access to an Oblivious Database Using Stateful Anonymous Credentials," Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography, pp. 501-520, 2009.
[5]J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious Transfer with Access Control," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 131-140, 2009.
[6]K.P.N. Puttaswamy, C. Kruegel, and B.Y. Zhao, "Silverline: Toward Data Confidentiality in Storage-Intensive Cloud Applications," Proc. Second ACM Symp. Cloud Computing (SOCC '11), pp. 10:1-10:13, 2011.
[7]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. VLDB, Sep. 2007, pp. 123–134.
[8]P. Samarati and S. D. C. di Vimercati, "Data protection in outsourcing scenarios: Issues and directions," in Proc. ASIACCS, Apr. 2010, pp. 1–14.
[9]S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proc. IEEE INFOCOM, Mar. 2010, pp. 534–542.
[10]C. Blundo, S. Cimato, S. D. C. Di Vimercati, A. D. Santis, S. Foresti,S. Paraboschi, et al., "Efficient key management for enforcing access control in outsourced scenarios," in SEC (IFIP), vol. 297. New York, NY, USA: Springer-Verlag, May 2009, pp. 364–375.
[11]V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. ACM Conf. Computer. Communication . Sec., Oct./Nov. 2006, pp. 89–98.
[12]C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in ASIACRYPT (Lecture Notes in Computer Science), vol. 2501. New York, NY, USA: Springer-Verlag, 2002, pp. 548–566.
[13]D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in EUROCRYPT (Lecture Notes in Computer Science), vol. 3494. New York, NY, USA: Springer Verlag, May 2005, pp. 440–456.
[14]A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. EUROCRYPT, 2005, pp. 457–473
[15]L. Zhou, V. Varadharajan, and M. Hitchens, "Enforcing role-based access control for secure data storage in the cloud," Computing . J., vol. 54, no. 13, pp. 1675–1687, Oct. 2011.
[16] Y. Zhu, D. Ma, C. Hu, and D. Huang, "How to use attribute-based encryption to implement role-based access control in the cloud," in Proc. Int. Workshop Sec. Cloud Computing ., 2013, pp. 33–40.
[17] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, "SiRiUS: Securing remote untrusted storage," in Proc. NDSS, 2003, pp. 1–15.
[18]G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in Proc. NDSS, Feb. 2005, pp. 29–43.
[19]X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute Based Proxy ReEncryption with Delegating Capabilities," Proc. Fourth Int'l Symp. Information, Computer, and Comm. Security (ASIACCS '09), pp. 276286, 2009.
[20]C.-K. Chu, J. Weng, S. Chow, J. Zhou, and R. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th Australasian Conf. Information Security and Privacy, pp. 327-342, 2009.
[21]Shiyuan Wang, "Secure and privacy-preserving database services in the cloud", ICDE, 2013, 2013 29th IEEE International Conference on Data Engineering (ICDE 2013), 2013 29th IEEE International Conference on Data Engineering (ICDE 2013) 2013, pp. 1268-1271, doi:10.1109/ICDE.2013.6544921
[22] M. J. Atallah, K. B. Frikken, and M. Blanton, "Dynamic and efficient key management for access

hierarchies," in Proc. ACM Conf. Computer Communication . Sec., Nov. 2005, pp. 190–202.
[23] Trust Enhanced Cryptographic Role-Based Access Control for Secure Cloud Data Storage , 10.1109/TIFS.2015.2455952, Nov 2015, pp.2381-2395
[24] Pervasive Computing and Applications (ICPCA), 2011 6th International Conference pp.330-334 Dec 2015.
[25] A 4D-Role Based Access Control Model for Multitenancy Cloud Platform, Mathematical Problems in Engineering Volume 2016 (2016), Article ID 2935638, 16 pages http://dx.doi.org/10.1155/2016/2935638

AUTHORS

Anirban Basu is an MTech in Electronics and a Masters and PhD in Computer Science with more than 30 years experience in Academia, advanced R&D, Software Industry, Consultancy and Corporate Training. Dr Basu is a recipient of a number of awards, which includes the prestigious 1985 Canadian Commonwealth Scholarship in Computer Science, Computer Engineering Division Gold Medals of The Institution of Engineers (India) .

Sunitha B S Working as Associate Professor in the Department of Information Science. Her Research interested includes Cluster Computing and Cloud Computing.