EFFICIENT FINE-GRAINED PRIVACY PRESERVING SYSTEM FOR PUBLIC CLOUD NETWORKS

K. Suganya, V. Geetha

Abstract—**An important problem in public clouds is how to selectively share documents based on fine-grained attribute-based access control policies (acps). An approach is to encrypt documents satisfying different policies with different keys using a public key cryptosystem such as attribute-based encryption, and/or proxy re-encryption. However, such an approach has some weaknesses: it cannot efficiently handle adding/revoking users or identity attributes, and policy changes; it requires to keeps multiple encrypted copies of the same documents; it incurs high computational costs. A direct application of a symmetric key cryptosystem, where users are grouped based on the policies they satisfy and unique keys are assigned to each group, also has similar weaknesses. We observe that, without utilizing public key cryptography and by allowing users to dynamically derive the symmetric keys at the time of decryption, one can address the above weaknesses. Based on this idea, we formalize a new key management scheme, called broadcast group key management (BGKM), and then give a secure construction of a BGKM scheme called ACV-BGKM. The idea is to give some secrets to users based on the identity attributes they have and later allow them to derive actual symmetric keys based on their secrets and some public information. A key advantage of the BGKM scheme is that adding users/revoking users or updating acps can be performed efficiently by updating only some public information. Using our BGKM construct, we propose an efficient approach for fine-grained encryption-based access control for documents stored in an untrusted cloud file storage.**

Index Terms—**Group key management, Privacy, Identity, Cloud computing, Policy, Encryption, Access control.**

I. INTRODUCTION

With the advent of technologies such as cloud computing, sharing data through a third-party cloud service provider has never been more economical and easier than now. However, such cloud providers cannot be trusted to protect the confidentiality of the data. In fact, data privacy and security issues have been major concerns for many organizations utilizing such services. Data often encode sensitive information and should be protected as mandated by various organizational policies and legal regulations. Encryption is a commonly adopted approach to protect the confidentiality of the data. Encryption alone, however, is not sufficient as organizations often have to enforce fine-grained access control on the data. Such control is often based on the attributes of users, referred to as *identity attributes*, such as the roles of users in the organization, projects on which users are working and so forth. These systems, in general, are called *attribute-based systems*. Therefore, an important requirement is to support fine-grained access control, based on policies specified using identity attributes, over encrypted data. With the involvement of the third-party cloud services, a crucial issue is that the identity attributes in the access control policies (acps) often reveal privacy-sensitive information about users and leak confidential information about the content. The confidentiality of the content and the privacy of the users are, thus, not fully protected if the identity attributes are not protected. Further, privacy, both individual as well as organizational, is considered a key requirement in all solutions, including cloud services, for digital identity management [2], [3], [4], [5]. Further, as insider threats [6] are one of the major sources of data theft and privacy breaches, identity attributes must be strongly protected even from accesses within organizations. With initiatives such as cloud computing the scope of insider threats is no longer limited to the organizational perimeter. Therefore, protecting the identity attributes of the users while enforcing attribute-based access control both within the organization as well as in the cloud is crucial. An approach to support fine-grained selective attribute based access control is to encrypt each content portion to which the same access control policy (or set of policies) applies with the same key, and then upload the encrypted content to the cloud. One approach to deliver the correct keys to the users based on the policies they satisfy is to use a hybrid solution where the keys are encrypted using a public key cryptosystem such as attribute-based encryption (ABE) and/or proxy re-encryption (PRE). However, such an approach has several weaknesses: it cannot efficiently handle adding/revoking users or identity attributes, and policy changes; it requires to keeps multiple encrypted copies of the same key; it incurs high computational cost. Therefore, a different approach is required. It is worth

K. Suganya, *Mailam Engineering College, Villupuram, Tamil Nadu, India.*
V. Geetha, *Mailam Engineering College, Villupuram, Tamil Nadu, India.*

ISSN: 2278 – 1323

noting that a simplistic group key management (GKM) scheme in which the content publisher directly delivers the symmetric keys to corresponding users has some major drawbacks with respect to user privacy and key management. On the one hand, user private information encoded in the user identity attributes is not protected in the simplistic approach. On the other the advent of technologies such as cloud computing, sharing data through a third-party cloud service provider has never been more economical and easier than now. However, such cloud providers cannot be trusted to protect the confidentiality of the data. In fact, data privacy and security issues have been major concerns for many organizations utilizing such services. Data often encode sensitive information and should be protected as mandated by various organizational policies and legal regulations. Encryption is a commonly adopted approach to protect the confidentiality of the data. Encryption alone, however, is not sufficient as organizations often have to enforce fine-grained access control on the data. Such control is often based on the attributes of users, referred to as *identity attributes*, such as the roles of users in the organization, projects on which users are working and so forth. These systems, in general, are called *attribute-based systems*. Therefore, an important requirement is to support fine-grained access control, based on policies specified using identity attributes, over encrypted data.

With the involvement of the third-party cloud services, a crucial issue is that the identity attributes in the access control policies (acps) often reveal privacy-sensitive information about users and leak confidential information about the content. The confidentiality of the content and the privacy of the users are, thus, not fully protected if the identity attributes are not protected. Further, privacy, both individual as well as organizational, is considered a key requirement in all solutions, including cloud services, for digital identity management [2], [3], [4], [5]. Further, as insider threats [6] are one of the major sources of data theft and privacy breaches, identity attributes must be strongly protected even from accesses within organizations. With initiatives such as cloud computing the scope of insider threats is no longer limited to the organizational perimeter. Therefore, protecting the identity attributes of the users while enforcing attribute-based access control both within the organization as well as in the cloud is crucial.

An approach to support fine-grained selective attribute based access control is to encrypt each content portion to which the same access control policy (or set of policies) applies with the same key, and then upload the encrypted content to the cloud. One approach to deliver the correct keys to the users based on the policies they satisfy is to use a hybrid solution where the keys are encrypted using a public key cryptosystem such as attribute-based encryption (ABE) and/or proxy re-encryption (PRE). However, such an approach has several weaknesses: it cannot efficiently handle adding/revoking users or identity attributes, and policy changes; it requires to keeps multiple encrypted copies of the same key; it incurs high computational cost. Therefore, a different approach is required.

It is worth noting that a simplistic group key management (GKM) scheme in which the content publisher directly delivers the symmetric keys to corresponding users has some major drawbacks with respect to user privacy and key management. On the one hand, user private information encoded in the user identity attributes is not protected in the simplistic approach. On the other hand, such a simplistic key management scheme does not scale well as the number of users becomes large and when multiple keys need to be distributed to multiple users. The goal of this paper is to develop an approach which does not have these shortcomings.

In recent days, cloud computing is used by many large and small organisations either directly or indirectly. Cloud service provides sharing of data in a more economical and easier way. The cloud services are divided in to three categories namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The Infrastructure as a Service (IaaS) provides the user with virtual infrastructures, such as servers, routers, switches and storage area. The Platform as a Service (PaaS) provides the user with development environment services where the user can create and run home-grown applications. The Software as a Service (SaaS) provides the user with access to already created applications that are operating in the cloud.This project work is based on Infrastructure as a Service (IaaS). Similar to cloud services, Cloud Computing Deployment models are divided in to four types namely, Public cloud, Private cloud, Hybrid cloud and Community cloud. Public cloud infrastructure is available to the general public and is owned by a third party Cloud Service Provider (CSP).The public cloud deployment model represents true cloud hosting in which, the services and infrastructure are provided to various clients. Private cloud infrastructure is operated solely for a single organization in whichthe CSP dedicates specific cloud services to that particular organization and no other clients are allowed to access the data. Hybrid cloud deployment model is the combination of private and public cloud which helps businesses to take advantage of secured applications and data hosting on a private cloud, while still enjoying cost benefits by keeping shared data and applications on the public cloud. Hence, the cloud migrates workloads between public and private hosting without any inconvenience to the users. In Community cloud deployment model, the cloud infrastructure is shared by several organizations with the same policy. This helps to further reduce costs as compared to a private cloud, as it is shared by larger groups.

The cloud deployment model which is used in this project work is a public cloud. In public cloud, the services are available to the general public and are controlled by data owner and a third party Cloud Service Provider (CSP). Google is a best example for public cloud. The services are provided to various clients by a vendor free of charge or on the basis of pay-per-user policy. This model greatly reduces the capital expenditure. In public cloud, since the data is public, many users can access the data located in the cloud service provider side. However, such cloud providers cannot be trusted to protect the confidentiality of the data placed by the data owner. Data privacy and security issues are the major concerns for many organizationsthat utilize the facility of accessing the data in from a public cloud. To provide the confidentiality of the data, as access control mechanism is to be implemented in public cloud networks.

In this project work, an efficient fine grained encryption based access control is proposed for documents stored in an untrusted public cloud networks. The users are allowed to access the documents to which they have access control. The access control is given to the users based on their identity attributes. To preserve the data, the identity of the user is to be protected. In this attribute based access control mechanism a user is able to decrypt the documents if and only if its identity attributes satisfy the data owner's access control policies. The data owner and the cloud learn nothing about the user's identity attributes. Thus, confidentiality of the data is provided by protecting the user's identity attributes. In order to implement the access control mechanism, computation efficient key management scheme is used.

II. RELATED WORKS

Access Control mechanisms are used for restricting unauthorized users from accessing the data. The widespread adaptation of internet standards, protocols and policies for information exchanged is laying a foundation for flexible granularity in information communication to provide services. There are many previous works on access control based security mechanisms used for wired and wireless networks. Among them, a Java based system to address the security issues of access control was developed, based on policy design for XML documents. This system supports the specification of policies at varying granularities and the trust level of users to enforce access control. Generally, a Role Based Access Control (RBAC) model consists of four basic components; a set of users, a set of roles, a set of permissions and a set of sessions. In a RBAC system, the user can be either a system user or an individual user. A role is one which is acquired by an individual on behalf of an organization through which the user will get a set of privileges to carry out a job function within a particular organization. Moreover, permission is a privilege given to the user in which the role is played by the user. When a user logs into a system, he/she establishes a new session using a key. During this session, the user can utilize the privileges of his role to perform various data manipulation activities on database tables and objects which are created for the organization. Roles have several advantages since roles represent an organizational function. A role based access control model can directly support an organizations security policy so that it helps the administration. The RBAC model is widely used for access control management, both in closed and open systems, where authorizations are specified with respect to roles and not with respect to individual users. Each user can have more than one privilege since they can play more than one role at a time. Based on these roles, privileges are assigned to each of the roles since managing few roles is much more efficient than managing most individual users. Because of its relevance, RBAC has been individually investigated by researchers. Though RBAC has been explored thoroughly, there are still significant application requirements which are not addressed by current RBAC models. To overcome this issue, a generality of RBAC model called the Action Status based Access Control (ASAC) was proposed.

A key feature of the ASAC model is that a decision on an agent's request to access resources is determined by considering the agent's ascribed status. In such a system, the agent's action status along with additional conditions of relevance is considered for processing the access request. An agent's attributed status together with the agent's action status gives a measure of the agent's overall status level. The agent's status level is used as the basis for determining authorized actions and thus is used in rendering a decision on the agent's access request. Another important criterion for security in distributed systems is the location constraints. Therefore, an access control system must not only consider temporal constraints but also the spatial constraints. In order to cope up with temporal and spatial constraints, the conventional RBAC model must be extended to specify temporal and spatial restrictions on permissions assigned to roles. Group access control can be achieved by encrypting the message using an encryption key with a large size. This key is dynamically generated for each session of the communication. Therefore, this dynamically generated key which is developed by using an effective key management scheme is known as the Session Key (SK) or Group Key (GK) that is shared by all legitimate users of a group to access a common data from the cloud server. This is necessary since the group membership in a multicast group is most likely to change dynamically since whenever a new user join or an existing member leave from the group, the encryption keys must be updated in order to prevent the leaving or joining user from accessing the data or messages from future or prior communications. The issues of establishing and updating the group keys have been addressed by various Group Key Management schemes present in the literature. Compared with all the existing key management schemes, the key management scheme proposed in this project work is a new work where the keys are not generated by a centralized authority or key server. Instead, a data owner is generating the private keys to each users from which it computes a common public key which is known as the Group Key (GK). After generating a common group key it encrypts the data using the Group Key (GK) and stores the encrypted data in the cloud servers. Moreover, in this project work, privacy of each user is also preserved by protecting their identity from data owner and cloud service provider side.

The process of generating, distributing and maintaining of keys are taken care by key management schemes. There are many key management schemes that are available in the literature. There are two types of key management schemes namely, centralized and distributed key management schemes that are used at present for providing security to multicast communication. In the centralized scheme, a trusted third party is used to control the activities of group management. These activities include member registration, key generation, key distribution and group management. Moreover, the trusted third party called GC is responsible for interacting with the group members and to control them in the centralized key management scheme. In contrast, the keys in a distributed key management scheme are computed and maintained with the coordination of group members. Distributed key management schemes are divided into two

types namely fully contributed key management and partially contributed key management schemes. In a fully distributed key management scheme, the users themselves contribute to form and distribute the key which helps to maintain the secrecy and group membership that provides security to the group communication. In a partial distributed key management scheme, both the users and the group centre are responsible for generating and maintaining the keys and group membership. In such a scenario, the group members are getting some amount of information from the group centre which is used by them to maintain the secrecy and group membership. The key management scheme developed in this project work is a centralized key management scheme that runs between data owner and cloud users.

The provision of access control facility in centralized key managements is a challenging task. This is due to the fact that the handling of key generation and distribution are more complex when the messages are distributed to a group of users from the cloud servers where number of users who join or leave the multicast group is more and dynamic. Therefore, in such a dynamic multicast group communication, it is necessary to allow the members to join or depart from the service at any time. When a new member joins into the service, it is the responsibility of the data owner to prevent the new member from having access to previous data in order to provide backward secrecy in the secure group communication. Similarly, when an existing group member leaves from any group, such a member should not have further access to data which is available only to the existing group members in the cloud server to achieve forward secrecy. In order to handle the issues of forward and backward secrecy, the keys are updated whenever a member joins or leaves to/from the multicast service. The Data owner takes the responsibility of generating a new group key after members join and leave operations are carried out. After a member joins the group or leaves from the group, the new group key is generated and is shared between data owner and group users in a secure way. That is, when a membership changes in the dynamic group, it computes a common public information and access control vector in the data owner side and it is sent as a broadcast message to all the users. After receiving the public information and access control vector, each user can compute a new group key by using their own secret key and the received public values. Thus, changing the group key securely after a member join or leave operation takes less computation and communication complexity in this project work.

Key Generation process in secure multicast communication is responsible for generating the random keys to be assigned privately to the registered users. This process also computes group keys with respect to the private keys related under the same sub group. An important issue in the maintenance of integrity in communication is to propose new techniques for generating a group key by the data owner and making the group members to derive the group key without showing the identity of individual members of the group. There are two types of techniques that are used for group key generation. In the first method, users are generating their own secret keys from which they compute a common group key which will be acting as a public key for a group of members. This method is called, distributed key management scheme. In the second method, a trusted third party called as data owner generates the group key and distributes them to the group members in a secure way. This method is called as centralized key management scheme. In both the schemes, several computations are necessary to compute the sub group and group keys. Moreover, both these schemes need storage for storing the public parameters and various key values used for computing the group key. In order to overcome the challenges on computational complexity and memory requirement, it is necessary to propose new key management schemes with reduced computation cost and memory requirement. Therefore, it is necessary to propose new computationally efficient technique that use simple mathematical functions, optimal number of multiplications and divisions in order to generate a group key effectively.

Key Distribution scheme in secure group communication is responsible for distributing the private keys and group keys to the registered users in the cloud network. Group keys can be distributed either by the data owner to the participating members or the members themselves will be distributing the keys generated by them which are necessary for computing the group key. In a centralized key management scheme, the group key is distributed by the data owner where as in the distributed approach any one of the group members can provide support for distributing the group key. This project work focuses on centralized key distribution schemes since real security challenges lie mostly on the effective design of a new centralized key distribution scheme. Even though, many design techniques have been developed by various researchers in the past, they are incurring more overhead with respect to increasing communication cost in the case of member join and leave operation. Moreover, all the existing key management schemes are not suitable to provide a group oriented service in the cloud network. Hence, it is necessary to propose new and effective centralized key distribution scheme suitable to provide a group oriented service in the cloud network that reduce the computation complexity.

In public cloud networks, the confidentiality of the data and the privacy of the users are not protected. In order to preserve the privacy of the user, the identity attribute of the user is to be protected. Many privacy preserving techniques are existing in the literature, but they are not efficient to protect the privacy of the user's identity attributes. To protect the privacy of the identity attributes of the users, two cryptographic techniques are used in this project work namely Pedersen commitment and OCBE protocols. In these techniques, the users are not directly submitting their attributes to data owner or to the cloud service provider to access the data. Instead, users are submitting their attributes to the token generator to get the identity tokens which contains commitment values, Pseudo Name and ID tag and the digital signature of these three. These identity tokens are sent to data owner to get the secret key values. The data owner is verifying the identity token by getting some information from the token generator. After verifying the identity token submitted by the cloud user, the data owner

generates the secret keys and distributes them for cloud user. Moreover, the data owner finds the access privileges of each users and encrypts the data accordingly to store it in the cloud server.

The cloud user can decrypt the data and view the data sent by the data owner if and only if the user satisfies the access control policy without showing the identity to the cloud server. Therefore, the data owner and the cloud service provider does not know anything about the user's identity attributes. Thus, the privacy of the users attribute is preserved in this project work.

In public cloud, the services are available to the general public and are controlled by data owner and a third party Cloud Service Provider (CSP). Since the data is public, many users can access the data located in the cloud service provider side. However, such cloud providers cannot be trusted to protect the confidentiality of the data placed by the data owner. Data privacy and security issues are the major concerns for many organizations that utilize the facility of accessing the data in from a public cloud. To provide the confidentiality of the data, access control mechanism is to be implemented in public cloud networks. In this project work, an efficient fine grained encryption based access control is proposed for documents stored in an untrusted public cloud networks. In order to implement the access control mechanism, computation efficient key management scheme is used. After a member joins the group or leaves from the group, the new group key is generated and is shared between data owner and group users in a secure way. Changing the group key securely after a member join or leave operation takes less computation and communication complexity in this project work.In this project work, privacy of each user is also preserved by protecting their identity from data owner and cloud service provider side.

Encryption and key management techniques are necessary for ensuring confidentiality of the data located in the public cloud. Since the unauthorized users do not possess the group key for that session, they cannot decrypt the information available in the public cloud. For IP multicast security, several key management schemes were proposed in the past. However, all these static key management schemes do not provide a solution for key change upon membership changes for providing effective security in the multicast group communication. Moreover, the existing key management schemes available in the literature, considered the access control issues for only wired and wireless networks. However, it is necessary to provide a facility for updating keys with respect to change in memberships dynamically in the cloud network and also to provide all the legitimate group members with the necessary level of access privileges. This helps to maintain forward and backward secrecy in the public cloud network.

In the past, many researchers have contributed and proposed various key management schemes for wired and wireless networks. Comparing with all the key management schemes that are existing in the literature, the key management schemes proposed in this thesis are different in many ways. First, key management scheme proposed and implemented in this research work for increasing the security and optimizing the key computation time of the system. In order to minimize the computation time, the cloud users are performing only one operation for finding the new group key which was computed by the data owner. In addition to this, this proposed key management scheme is suitable for cloud network.

Second, the proposed key distribution and key management provides privacy preserving methods there by it preserves the privacy of the user for improving the users secrecy. Finally, a batch key updating algorithm also called as batch rekeying has been proposed in this thesis work which reduces the number of rekeying operations required for batch leave or join operations. Therefore, the key management scheme proposed in this thesis work has been designed in such a way that they reduce the computational complexity in all cases with the exception of few cases for increasing the security. In short, this thesis work provides new algorithms which are computation, communication and memory efficient in comparison with other existing works.

IV. CONCLUSION

In this project work, a privacy preserving algorithm is implemented to preserve privacy of the each user who is accessing the data from the public cloud. In addition to this, a broadcast key management is also implemented in this project work to provide access control vector for all the users who belong to a particular group. The access control vector is used to compute a common group key to perform the decryption operation in the user's side. The implemented algorithm is computation and memory efficient in the cloud user's side. Even though implemented algorithm is computation efficient, it is used only for single user join and leave operation in this project. This work can also be extended to batch rekeying operations when a group of users joins and leaves the group at a time. Moreover, the broadcast key management used in this project works user public information(PI) which consists of all users public key and access control vector to compute the group keys . This would increase the communication complexity when the numbers of users are high in the multicast group for example, when a group consists of 1000 users, the data owner has to send all the 1000 users. Therefore, the communication complexity of his broadcast key management scheme is O(n+1) where 'n' is the number of users in the group and 1 is the Access control vector size. The communication complexity can also be reduced by developing a new communication, computation and storage efficient broadcast key management scheme in the future works moreover, if the Access control vector is lost(or)corrupts during the transmission time, then all the users cannot find the group key which also a challenging issue in the implemented broadcast key management scheme. In order to avoid this, a reliability approach can also be integrated into this implement key management algorithm.

REFERENCES

[1] AbdulhadiShoufan, Sorin, A. and Huss, "High-Performance Rekeying Processor Architecture for Group Key Management", IEEE Transactions on Computers, Vol. 58, No.10, pp.1421-1434, 2009.

[2] Adrian Perrig, Dawn Song, J.D. and Tygar "ELK A New Protocol for Efficient Large-Group Key Distribution", Proceedings of IEEE Symposium on Security and Privacy Symposium, pp. 247–262, 2001.

[3] BaihuaZheng, Wang-Chien Lee, Peng Liu, DikLun Lee and Xuhua Ding, "Tuning On-Air Signatures for Balancing Performance and Confidentiality", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 12, pp. 1783-1797, 2009.

[4] BezawadaBruhadeshwar, Sandeep S. Kulkarni and Alex X. Liu, "Symmetric Key Approaches to Securing BGP—A Little Bit Trust is Enough", IEEE Transactions on Parallel and Distributed Systems,Vol. 22, No. 9, pp. 1536-1549, 2011a.

[5] Bezawada Bruhadeshwar, Sandeep, S. and Kulkarni, "Balancing Revocation and Storage Trade-offs in Secure Group Communication", IEEE Transactions on Dependable and Secure Computing, Vol. 8, No.1, pp. 58-73, 2011b.

[6] Bin Liu, Yurong Jiang, Fei Sha, Ramesh Govindan, "Cloud-Enabled Privacy-Preserving Collaborative Learning forMobile Sensing" , ACM conference on embedded networked sensor systems,2012.

[7] Brian Zhang, X., Lam, S., Young Lee, D. and Richard Yang,Y. "Protocol Design for Scalable and Reliable Group Rekeying", IEEE /ACM Transactions on Networking, Vol. 11, No.6, pp. 908-922, 2003.

[8] Chih-Lin Hu and Ming-Syan Chen**,** "Adaptive Information Dissemination: An Extended Wireless Data Broadcasting Scheme with Loan-Based Feedback Control", IEEE Transactions on Mobile Computing, Vol. 2, No. 4, pp. 322-336, 2003.

[9] Chun-I Fan, Ling-Ying Huang and Pei-HsiuHo**,** "Anonymous Multireceiver Identity-Based Encryption"**,** IEEE Transactions on Computers ,Vol. 59, No. 9, pp. 1239-1249, 2010.

[10] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage" , IEEE transactions on computers, vol.62 No.2, pp.362-375, 2013

[11] David, A., McGrew and Alan T. Sherman, "Key Establishment in Large Dynamic Groups using One-Way Function Trees", IEEE Transactions on Software Engineering, Vol. 29, No. 5, pp. 444-458, 2003.

[12] Dirk Westhoff, Joao Girao and Mithun Acharya, "Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation", IEEE Transactions on Mobile Computing, Vol. 5, No. 10, pp. 1417-1431, 2006.

[13] Donggang, L., Peng Ning and Rongfang Li, "Establishing Pairwise Keys in Distributed Sensor Networks", ACM Transactions on Information and System security, Vol. 8, No. 1, pp. 41-77, 2005.

[14] Dong-Hyun Je, Jun-Sik Lee, Yongsuk Park and Seung-Woo Seo, "Computation-and-Storage Efficient Key Tree Management Protocol for Secure Multicast Communications", Elsevier, Computer Communications, Vol. 33, No. 6, pp. 136-148, 2010.

[15] Fei, Z., Ammar, M.H., Kamel, I. and Mukherjee, S. "An Active Buffer Management Technique for Providing Interactive Functions in Broadcast Video-On-Demand Systems", IEEE Transaction Multimedia, Vol. 7, No. 5, pp. 942-950, 2005.

[16] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux and Antonio Lioy, "On the Performance of Secure Vehicular Communication Systems", IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 6, pp. 898-912, 2011**.**

[17] Goshi, J. and Ladner R.E. "Algorithms for Dynamic Multicast Key Distribution Trees," Procedings of ACM Symposium on Principles of Distributed Computing, pp. 243-251, 2003.

[18] Haibin Lu, "A Novel High-Order Tree for Secure Multicast Key Management", IEEE Transactions onComputers, Vol. 54, No.2, pp. 214-224, 2005.

[19] Harney,H., Harder,E., "Logical key hierarchy protocol," Internat Draft, IETF, Expired in August 1999.

[20] Hai-Tao Xie, Zong-Kai Yang, Yu-Ming Wang and Wen-Qing Cheng, "A M-dimensional Sphere Multicast Rekeying Scheme", Communications and Mobile Computing International Conference, Vol. 3, pp. 418-422, 2009.s

[21] Hock Desmond Ng, W., Howarth, M., Sun, Z. and Cruickshdank. H. "Dynamic Balanced Key Tree Management for Secure Multicast Communications", IEEE Transactions on Computers, Vol. 56, No. 5, pp. 590-605, 2007.

[22] Hongsong Shi and Mingxing He, "A Communication-Efficient Key Agreement Protocol in Ad Hoc Networks", IEEE International Conference on Wireless Networks, Communications and Mobile Computing , China, pp. 285-291, 2005.

[23] Hui Chen and Yang Xiao, "On-Bound Selection Cache Replacement Policy for Wireless Data Access", IEEE Transactions on Computers, Vol. 56, No. 12, pp. 1597-1611, 2007.

[24] Jian Wang, Yan Zhao, Shuo Jiang, Jiajin Le, "Privacy preserving system to support cloud computing", human system interactions 3rd conference, 2010.

[25] Jin-Hee, C., Ing-Ray, C. and Mohamed, E. "On Optimal Batch Rekeying for Secure Group Communications in Wireless Networks," Springer, Journal on Wireless Networks, Vol. 14, No.6, pp. 915-927, 2008.

K. Suganya has received her B.E.(CSE) degree in the year 2013. At present She is pursuing M.E. (CSE) in Mailam Engineering College, Villupuram, Tamil Nadu, India. She has published 4 papers in National conferences.Her research interests lies in the areas of Mobile Computing, Data Mining, Cloud Computing and Software Engineering.

V. Geetha Completed her B.E. (CSE) degree in the year 2005, M. Tech(CSE) degree in the year 2012.Currently she is working as Assistant professor in Computer Science and Engineering at Mailam Engineering College, Villupuram, Tamil Nadu, India. Her research interests lies in the areas of Data mining, Software engineering and Cloud Computing. She has published 2 papers in International conferences and 4 papers in National conferences. She attended many workshops & National seminars in various technologies and also attended Faculty Development Programme.

