INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

DAC: AN EFFICIENT SECURE APPROCH TO MAINTAIN CONFIDENTIALITY OF DATA

Shaik Habeeb Sultana[1], K.Sreenivasulu[2] (Ph.D.)

[1]M.Tech Student Department of CSE, Madina Engineering College, Kadapa, Andhra Pradesh, India
Email:habeeb553shaik@gmail.com[1]
[2] Prof, Head of Department of CSE, Madina Engineering College, Kadapa, AndhraPradesh, India
Email:sreenu.kutala@gmail.com[2]

Abstract: - Most of the cloud storages are used for data storage purpose on that we have a one entity called public cloud. Generally a public cloud prefers that storing the end user content information. By considering that the data owners enforce a fine-grained access control on confidential data hosting on the public clouds. Eventually such approaches are in charge of encrypting the data before uploading similarly if any user's policies or authorization policies changes then need to re-encrypt the data. Thus the data owners incur high communication and computation costs. A better approach should delegate the enforcement of fine-grained access control to the cloud, so to minimize the overhead at the data owners, while assuring data confidentiality from the cloud. We propose an approach a delegated access control, an efficient secure approach to maintain confidentiality of data, based on two layers of encryption that addresses such requirement. First the data owners performs coarse-grained encryption whereas cloud performs a fine-grained encryption A challenging issue is how to decompose access control policies (ACPs) such that the two layer encryption can be performed. We show that this problem is NP-complete and propose novel optimization algorithms. We utilize an efficient group key management scheme that supports expressive ACPs.

Keywords*:* Cloud computing, policy management, novel optimization, data encryption, DAC

1. Introduction

A delegated privacy represents major concerns in the adoption of cloud technologies for data storage. An approach to mitigate these concerns is the use of encryption. However, whereas 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 policies (ACPs).

Many organizations have today ACPs regulating which users can access which data; these ACPs are often expressed in terms of the properties of the users, referred to as identity attributes, using access control languages such as XACML. Such an approach, referred to as attribute-based access control (ABAC), supports fine-grained access control which is crucial for high-assurance data security and privacy [1][3].

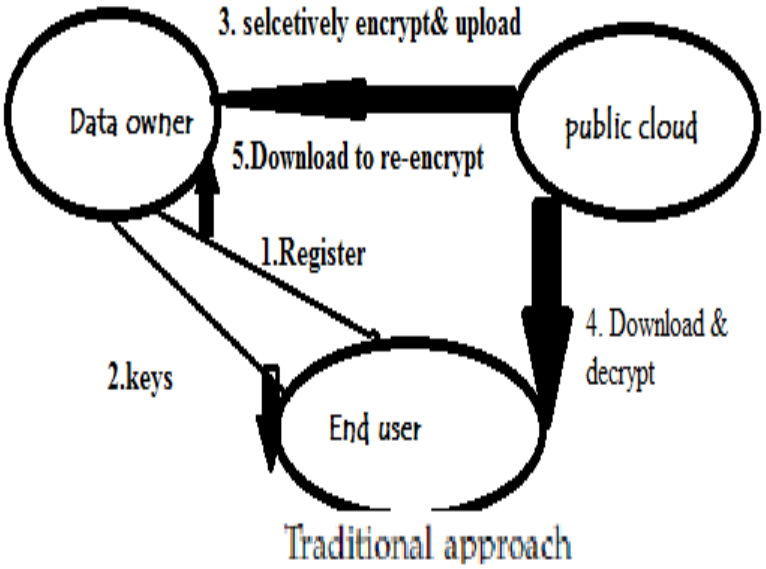

Figure 1: Traditional Approach

Supporting ABAC over encrypted data is a critical requirement in order to utilize cloud storage services for selective data sharing among different users. Notice that often user identity attributes encode private information and should thus is strongly protected from the cloud, very much as the data themselves.

By considering that the data owners enforce a fine-grained access control on confidential data hosting on the public clouds. Eventually such approaches are in charge of encrypting the data before uploading similarly if any user's policies or authorization policies changes then need to re-encrypt the data. Thus the data owners incur high communication and computation costs. A better approach should delegate the enforcement of fine-grained access control to the cloud, so to minimize the overhead at the data owners, while assuring data confidentiality from the cloud. We propose an approach a delegated access control, based on two layers of encryption that addresses such requirement as shown in the figure1. First the data owners performs coarse-grained encryption whereas cloud performs a fine-grained encryption A challenging issue is how to decompose access control policies (ACPs) such that the two layer encryption can be performed. We show that this problem is NP-complete and propose novel optimization algorithms. We utilize an efficient group key management scheme that supports expressive ACPs. [5], [6], [7].

2. Cloud Computing

To understand IT as a service it is best to start with an understanding of the cloud models implemented today. Cloud services can be delivered in one of multiple formats. Often, and IT department will start with a private cloud environment and perhaps focus on deploying virtual servers. As shown in figure 2 .This is commonly referred to as infrastructure as a service and represents one of the flavours of IT as a service that is available within the world of cloud computing. Since this is only a single example of the possible cloud models, IT professionals must become familiar with a variety of cloud standards so that they can select appropriately based on the needs of the enterprise. It is common to divide cloud computing into three categories:

2.1 Infrastructure as a service (IaaS), which provides flexible ways to create, use **and** manage virtual machines (VMs). Required to support applications

2.2 Platform as a service (PaaS), focused on providing the higher-level capabilities—more than just VMs—required to support applications**.**

2.3 Software as a service (SaaS), the applications that provide business value for users.

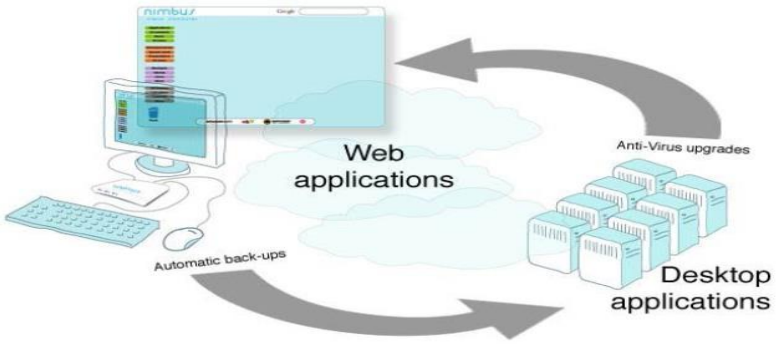

Figure 2: Deployment Models

For each cloud computing category there are additional decisions regarding the type of cloud chosen. The type of cloud that is selected determines the placement and usage model of the physical infrastructure that is being removed from the customer's data centre world. Essentially, the cloud computing deployment model describes where the software runs and includes the following options:

2.4 A private cloud is a set of standardized computing resources that is dedicated to an organization, usually on-premises in the organization's data centre. It works with the current capital investment and delivers the new functions as a service.

2.5 A hosted private cloud has a dedicated infrastructure hosted by a third, inaccessible to other organizations.

2.6 A public cloud consists of computing resources hosted externally but shared with other organizations and dynamically provisioned and billed on a utility basis — the customer will pay for what is used as they use it .Keeping these categories in mind, the next sections of the whitepaper discuss the service models and explore the roles and skills IT professionals and developers need to invest in for each of them.

3. Policy Management

Obviously once we start adding in public cloud hosting; firewall rule management can become even more unruly. If we continue to apply the above hierarchical structure, management can quickly turn into a nightmare. This is because we've increased the number of points on the first level of structure, namely the number of individual firewalls that must be managed. The complexity quickly runs down the structure, increasing the number of unique rules that must be managed.

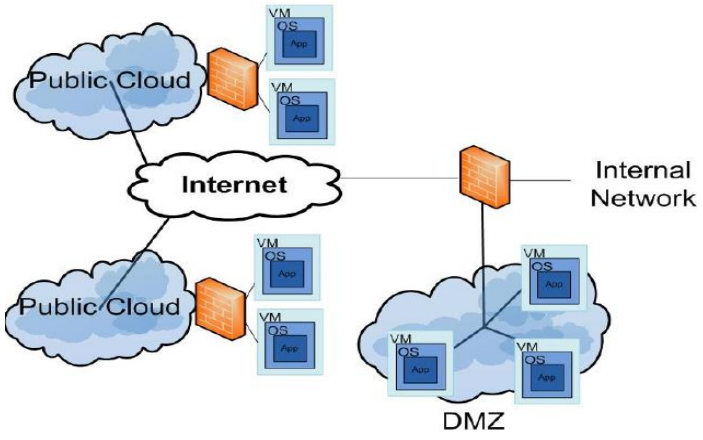

Figure 3: Incremental resources to manage an additional security for their cloud servers

As shown in the above figure 3 customers have scarce incremental resources to manage an additional security system for their cloud servers. And with short-term internal projects being assigned to public clouds without proper IT security protocol, the once robust walls of security can quickly become porous, while breach potential is increased exponentially. Public Cloud Server Security Suite leverages the power and scalability of PO software to identify, manage, and report across the entire server infrastructure: physical, virtual, and cloud. PO software provides a single "pane of glass" visibility into those shadows IT instances for highly scalable,

flexible, and automated centralized management and enforcement of security management policies to identify. If the access control related encryptions somehow delegated to the Cloud, the Owner 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 Owner has to initially encrypt the data and upload the encrypted data to the cloud. Therefore, in order for the Cloud to allow enforcing authorization policies through encryption and avoiding re-encryption by the Owner, 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 Owner. The OEL is for fine-grained authorization for controlling accesses to the data by the users and is generated by the Cloud [12].

An important issue in the TLE approach is how to distribute the encryptions between the Owner and the Cloud. There are two possible extremes. The first approach is for the Owner 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 Owner and the Cloud to perform the complete access control related encryption twice. The first approach has the least overhead for the Owner, but it has the highest information exposure risk due to collusions between Users and the Cloud. Further, IEL updates require to re-encrypting all data items. The second approach has the least information exposure risk due to collusions, but it has the highest overhead on the 3Owner as the Owner has to perform the same task initially as in the SLE approach and, further, needs to manage all identity attributes. An alternative solution is based on decomposing ACPs so that the information exposure risk and key management overhead are balanced. The problem is then how to decompose the ACPs such that the Owner has to manage the minimum number 7 of attributes while delegating as much access control enforcement as possible to the Cloud without allowing it to decrypt the data [5]. In what follow we propose such an approach to decompose and we also show that the policy decomposition problem is hard.

```
PolicyId=P, RuleCombiningAlgorithm = deny-override
  ⟨RuleId=r1 Effect=Permit⟩
    ⟨Target⟩
      ⟨Subject ProjectName = "SecretCrypto" ⟩
      ⟨Action Action= "Buy"⟩
    ⟨/Target⟩
    ⟨Condition ProjectRole = "PI" and
               ProjectLevel = "High"⟩
  ⟨/Rule⟩
  ⟨RuleId=r2 Effect=Deny⟩
    ⟨Target⟩
      ⟨Subject ProjectName = "SecretCrypto"⟩
      ⟨Action Action= "Buy"⟩
    ⟨/Target⟩
    ⟨Condition ProjectLevel = "High" and
               Funding < 100000 ⟩
  ⟨/Rule⟩
```

Figure 4: Group Policy

In figure 4 a sample algorithm that represents the group of policy decomposition.

4. Novel Optimization

Formally, NP-complete is a notion for so called recognition (or decision) problems, i.e., problems defined by a question for which the only two possible answers are a YES or a NO. It is defined with respect to polynomial reductions. From Nemhauser and Wolsey "X □ NP is said to be NP -complete if all problems in NP can be polynomials reduced to X." The first set of problems in this class is shown in Figure 1.

NP-hard problems are usually optimization problems whose recognition version is NP-complete. For example the TSP-optimization is NP-hard because its TSP-recognition version is NP-complete. Nemhauser and Wolsey say "A problem is NP-hard if there is an NP-complete problem that can be polynomials reduced to it." Thus if a problem is NP-hard it is at least as difficult as any NP-complete problem.

An efficient optimization algorithm is presented for the problems with hard to evaluate objective functions. It uses the reinforcement learning principle to determine the particle move in search for the optimum process. A model of successful actions is build and future actions are based on past experience. The step increment combines exploitation of the known search path and exploration for the improved search direction. The algorithm does not require any prior knowledge of the objective function, nor does it require any characteristics of such function. It is simple, intuitive and easy to implement and tune. The optimization

algorithm was tested using several multi-variable functions and compared with other widely used random search optimization algorithms. Furthermore, the training of a multi-layer perception, to find a set of optimized weights, is treated as an optimization problem. The optimized multi-layer perception was applied to Iris database classification. Finally, the algorithm is used in image recognition to find a familiar object with retina sampling and micro-saccades.

4.1 Basic Search Procedure:

A *N*-variable optimization objective function *V = f* (*p*1*; p*2*; :::; pN*) (*p*1*; p*2*; :::; pN;V 2* Â1) Could have several local minima and several global minima *Vopt*1*; :::;VoptN*. It is desired that the search process, initiated from a random point, finds a path to the global optimum point. Unlike particle swarm optimization, this process can be performed with a single search particle that learns how to find its way to the optimum point. It does not require the cooperation among a group of particles, although implementing the cooperation among several search particles may further enhance the search process in this method. At each point of the search, the search particle intends to find a new location with a better value within a searching range around it and then determines the direction and the step size for the next move. It tries to reach the optimum by exploring weighted random search of each variable (coordinate). The step size of search in each variable is randomly generated with its own probability density function. These functions are gradually learned during the search process. It is expected that at the later stage of search, the probability density functions are approximated for each variable. Then the stochastically randomized path to the minimum point of the function from the start point is learned

```
BEGIN RLO
Initialize: start from a random point V(p1,....pN)
   make changes to each coordinate p1'=p1+dp1, ..., pN'=pN+dpN

While (termination condition is not met – the step size or change of V is not too small) do
      For (j=1 to M trials)
            While (found new points have no better value than the current position)
                  move to new point Vnew(p1',....pN')
                  use historical data of dp1,....,dpN to do function approximation
                  use the function to predict dp1,....,dpN, Δdp1,....,ΔdpN
                  make changes to each coordinate p1'=p1+dp1, ..., pN'=pN+dpN
            End While
      End For
      change the center and the radius Δdp1,....,ΔdpN of searching area
End While
END
```

Figure 5: Basic Search Procedure Algorithm

5. Data Encryption

Data encryption was introduced to solve the problem of how to efficiently encrypt a message and broadcast it to a subset of the users in a system. The subset of users can change dynamically. In the broadcast encryption literature, these users are called privileged and the non-authorized users revoked. We denote the set of users by U, the set of revoked users R. The set of privileged users is thus U\R. We set N = |U| and r = |R|. While all users can get the encrypted message, only the privileged users can decrypt it. The simplest broadcast encryption scheme [9],[10].Simply consists of encrypting a message for each privileged user separately and the broadcasting all the encrypted messages[9]. Obviously, this scheme is very inefficient as the message length is prohibitively large (O (N−r)). Better data encryption schemes aim to reduce the following parameters:

- The processing time at the server to encrypt the message for the privileged users.
- The processing time at privileged users to decrypt messages.
- The broadcast message size.
- The storage size at both the server and privileged user

There are two approaches to broadcast encryption [9]. The first approach assumes that users are state-full meaning that the keys given to users can be updated when a new user is added or an existing user is revoked. The second approach assumes that users are stateless meaning that the keys given to users cannot be updated and can only be discarded. We consider only the latter approach since in the outsourced scenarios the keys initially given to users are difficult to update and, therefore, remain unchanged.

Setup (ℓ, N): The server constructs a binary tree Λ where there are at least N leaf nodes. Each node in Λ is either assigned a unique key whose length is decided by the security parameter ℓ, or can computationally derive a unique key. The user ui, i =1,2,⋯ ,N, is assigned the i^{th} leaf node.

Get Sec Keys (ui**)**: The server gives all the keys assigned to ui in Λ.

Get Cover (U\R**)**: Given the privileged user set U\R, the server outputs the cover C.

Broadcast (M**,** C**)**: The server generates a session key K and encrypts the message M with K and encrypts K with each key in the cover C.

Key-Der (ui**,** C**)**: The user ui identifies its subset in the cover C, outputs the key that decrypts the session key.

Decrypt(C**,** K**)**: It decrypts the encrypted message C with the key K, to output the message M.

Having defined the algorithms, we give a high-level description of the basic subset-cover technique. In the basic scheme, N users are organized as the leaves of a balanced binary tree of height log-N. A unique secret is assigned to each vertex in the tree is given log-N secrets that correspond to the vertices along the path from its leaf node to the root node. In order to provide forward secrecy when a single user is revoked, the updated tree is described by log-N sub trees formed after removing all the vertices along the path from the user leaf node to the root node. To rekey, the server uses the log-N secrets corresponding to the roots of these sub-trees. Many improved subset-cover based broadcast encryption algorithms have been proposed. In this work, we consider the complete sub-tree algorithm .The complete sub-tree algorithm improves the basic technique for simultaneously revoking r users and describing the privileged users using r log (N/r) subsets. Each user stores log-N keys.

6. DAC

Access control is the means by which administrators can control, or delegate, the ability of other users to manipulate objects in Active Directory and also to perform actions on domain controllers and file servers. Understanding the access-control model in Active Directory is essential to being able to delegate administration. This section provides an overview of the access control model in Active Directory and describes all relevant aspects of access control that are required to delegate administrative authority. Access control involves three components:

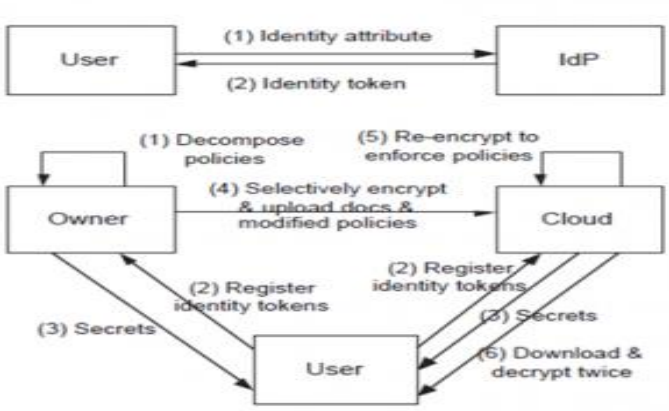

Figure 6: Delegated Access control model

The security credentials of the user attempting to access a resource, Authorization data that protects the resource that is being accessed. An access check that evaluates whether or not the requested access can be granted When a user (or a process that is running on behalf of the user) attempts to perform a low-level operation on a securable object, the operation being attempted is subject to an access check. The access check takes into account the user security credentials and the authorization data on the object on which the low-level operation is being requested to determine the abilities of the user in relation to the respective object. If the access check determines that the security credentials of the user requesting the operation and the authorization data on the target object provides sufficient permissions to execute the operation, the operation succeeds. If the user has insufficient permissions to execute the operation that is being requested, the request fails.

The act of delegating Active Directory administrative responsibilities involves identifying the low-level operation that corresponds to the administrative task and the specific data on which it is being performed, and then appropriately modifying authorization settings that protect the data [12].

7. RESULT ANALYSIS

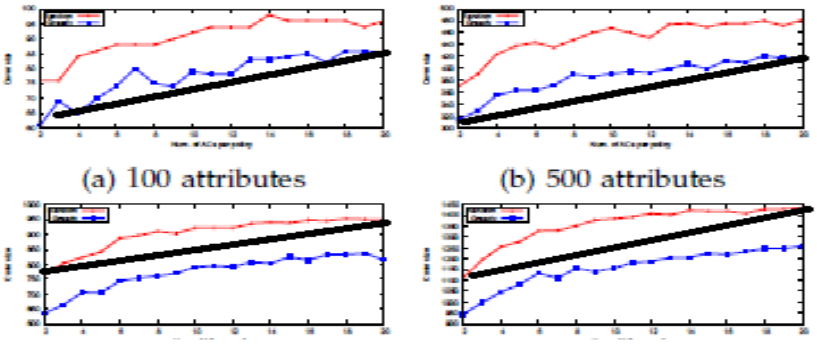

Figure 7: Size of ACCs for different number of ACs

The above figure 7 shows the attribute condition size that covers. That is, the data owner enforce the no. of attributes conditions for system having 100 and 500 ACs as number of attribute condition increases the per policy also increases**.**

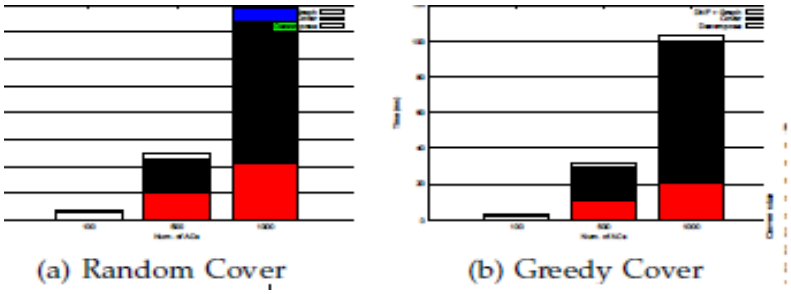

Figure 8: Time Break Down For Decomposing Policies

Figure 8 shows the break down time for decomposing the attribute control policies. The greedy policy cover algorithm performs better than the random cover algorithm. As a result for each policy the attribute constraints increases the cover size also increases.

We first present experimental results concerning the policy decomposition algorithms. We then present an experimental comparison between the SLE and TLE approaches. Adjacency list representation is used to construct policy graphs used in the two approximation algorithms for finding a near optimal attribute condition cover. We utilized the AB-GKM scheme with the subset cover optimization.

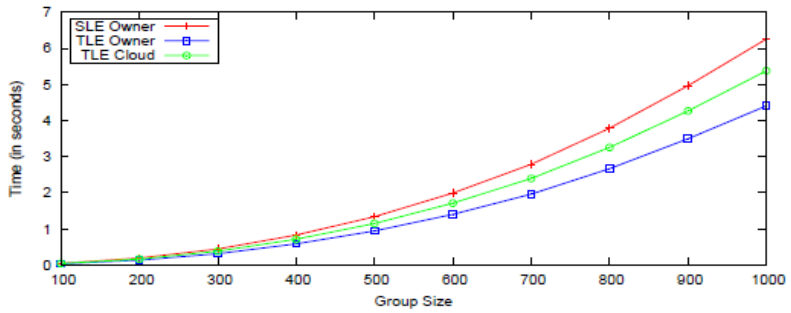

Figure 9: Average time to generate keys for the two approache*s*

Figure 9 reports the average time spent to execute the AB-GKM::KeyGen with SLE and TLE approaches for different group sizes. We set the number of attribute conditions to 1000 and the maximum number of attribute conditions per policy to 5. We utilize the greedy algorithm to find the attribute condition cover. As seen in the diagram, the running time at the Owner in the SLE approach is higher since the Owner has to enforce all the attribute conditions. Since the TLE approach divides the enforcement cost between the Owner and the Cloud, the running time at the Owner is lower compared to the SLE approach. The running time at the Cloud in the TLE approach is higher than that at the Owner since the Cloud performs fine grained encryption whereas the Owner only performs coarse grained encryption. As shown in Figure 10, a similar pattern is observed in the AB-GKM::KeyDer as well [11].

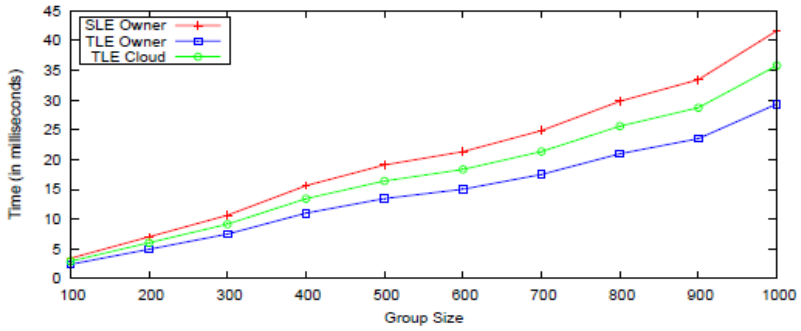

Figure 10: Average time to derive keys for the two approaches

7. Conclusion

The approaches to enforce ACPs on outsourced data using selective encryption require organizations to manage all keys and encryptions and upload the encrypted data to the remote storage. Such approaches incur high communication and computation cost to manage keys and encryptions whenever user credentials or organizational authorization policies/data change. In this paper, we proposed a two layer encryption based approach to solve this problem by delegating as much of the access control enforcement responsibilities as possible to the Cloud while minimizing the information exposure risks due to colluding Users and Cloud. A key problem in this regard is how to decompose ACPs so that the Owner has to handle a minimum number of attribute conditions while hiding the content from the Cloud. We showed that the policy decomposition problem is NP-Complete and provided approximation algorithms. Based on the decomposed ACPs, we proposed novel approach to privacy preserving fine-grained delegated access control to data in public clouds. Our approach is based on a privacy-preserving attribute based key management scheme that protects the privacy of users while enforcing attributes based ACPs. As the experimental results show, decomposing the ACPs and utilizing the two layer of encryption reduce the overhead at the Owner.

8. Future Enhancement:

As future work, we plan to investigate the alternative choices for the TLE approach further. We also plan to further reduce the computational cost by exploiting partial relationships among ACPs.

9. REFERENCES

[1] M. Nabeel and E. Bertino, “Privacy preserving delegated access control in the storage as a service model,” in EEE International Conference on Information Reuse and Integration (IRI), 2012.
[2] 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 Proceedings of the 33rd International Conference on Very Large Data Bases, ser. VLDB ’07. VLDB Endowment, 2007, pp. 123–134.
[3] E. Bertino and E. Ferrari, “Secure and selective dissemination of XML documents,” ACM Trans. Inf. Syst. Secur., vol. 5, no. 3,pp. 290–331, 2002.
[4] X. Zhang, S. Oh and R. Sandhu, “PBDM: a flexible delegation model n RBAC,” In proceedings of the eighth ACM symposium on Access control models and technologies (SACMAT), New York, NY, USA, pp. 149–157, 2003.
[5] N. Shang, M. Nabeel, F. Paci, and E. Bertino, “A privacy preserving approach to policy-based content dissemination,” in ICDE ’10: Proceedings of the 2010 IEEE 26th International Conference on Data Engineering, 2010.

[6] M. Nabeel, E. Bertino, M. Kantarcioglu, and B. M. Thuraisingham, "Towards privacy preserving access control in the cloud," in Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, ser. CollaborateCom '11, 2011, pp. 172–180.
[7] M. Nabeel, N. Shang, and E. Bertino, "Privacy preserving policy based content sharing in public clouds," IEEE Transactions on Knowledge and Data Engineering, 2012.
[8] M. Nabeel and E. Bertino, "Towards attribute based group key management," in Proceedings of the 18th ACM conference on Computer and communications security, Chicago, Illinois, USA,2011.
[9] A. Fiat and M. Naor, "Broadcast encryption," in Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, ser. CRYPTO '93. London, UK: Springer-Verlag, 1994, pp. 480–491.
[10] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, ser. CRYPTO '01. London, UK: Springer-Verlag, 2001, pp. 41–62.
[11] Syed Yasmeen and M. Naveen Kumar "An Efficient and Secured Storage Delegated Access Control to Maintain Confidentiality of Data", International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Special Issue 4, September 2014, ISSN 2320-9801
[12] M. Nabeel and E. Bertino, "Privacy Preserving Access Control for Third-Party Data Management System" In ACM Symposium on Access Control Models and Technologies (SACMAT), 2012.

10. BIOGRAPHY

Shaik Habeeb Sultana is a student in Master of Technology in the Department of Computer Science and Engineering, Madina Engineering College, Kadapa, Andhra Pradesh, India

K.Sreenivasulu is a Head of the Department of Computer Science and Engineering, Madina Engineering College, Kadapa, Andhra Pradesh, India.