ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Secrecy-Preserving Coarse grained and Fine-Grained Access Control in Public Clouds

Vinay Kumar Kommu
PG Scholar,
Department of CSE,
Sri indu College of Engineering & Technology.

T.Charan Singh
Assistant professor,
Department of CSE,
Sri indu College of Engineering & Technology.

Dr.Ch.G.V.N.Prasad
Professor & HOD,
Department of CSE,
Sri indu College of Engineering & Technology.

Abstract:

Current approaches to enforce fine-grained access control on confidential data hosted in the cloud are based on fine-grained encryption of the data. Under such approaches, data owners are in charge of encrypting the data before uploading them on the cloud and re-encrypting the data whenever user credentials change. Data owners thus 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, based on two layers of encryption that addresses such requirement. Under our approach, the data owner performs a coarse-grained encryption, whereas the cloud performs a fine-grained encryption on top of the owner encrypted data. 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. Our system assures the confidentiality of the data and preserves the privacy of users from the cloud while delegating most of the access control enforcement to the cloud.

Keywords: Privacy, Identity, Cloud Computing, Policy Decomposition, Encryption, Access Control.

I. INTRODUCTION:

Security and privacy represent 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. 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.

Approaches based on encryption have been proposed for fine-grained access control over encrypted data [3], [4]. As shown in Fig.1, those approaches group data items based on ACPs and encrypt each group with a different symmetric key. Users then are given only the keys for the data items they are allowed to access. Extensions to reduce the number of keys that need to be distributed to the users have been proposed exploiting hierarchical and other relationships among data items. Such approaches however have several limitations:

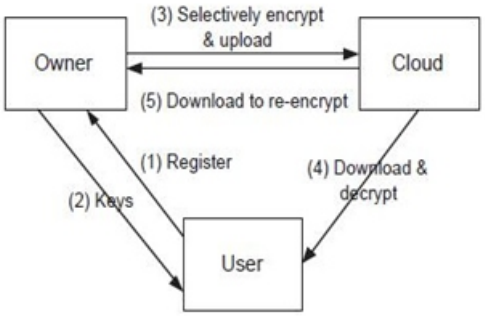


Fig.1. Traditional approach.

•In order to issue the new keys to the users, the data owner needs to establish private communication channels with the users.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

•The privacy of the identity attributes of the users is not taken into account. Therefore the cloud can learn sensitive information about the users and their organization.

•They are either unable or inefficient in supporting fine-grained ABAC policies.

Recently proposed approaches based on broadcast key management schemes [5], [6], [7] address some of the above limitations. We refer to these approaches as single layer encryption (SLE) approaches, since, like previous approaches, they require the data owner to enforce access control through encryption performed at the data owner. However, unlike previous approaches, SLE assures the privacy of the users and supports fine-grained ACPs. However, while SLE addresses some limitations of previous approaches, it still requires the data owner to enforce all the ACPs by fine-grained encryption, both initially and subsequently after users are added/ revoked. All these encryption activities have to be performed at the owner that thus incurs high communication and computation cost. For example, if a user is revoked, the owner must download from the cloud the data affected by this change, generate a new encryption key, re-encrypt the downloaded data with the new key, and then upload the re-encrypted data to the cloud. In this paper, we propose a new approach to address this shortcoming.

The approach is based on two layers of encryption applied to each data item uploaded to the cloud. Under this approach, referred to as two layer encryption (TLE), the data owner performs a coarse grained encryption over the data in order to assure the confidentiality of the data from the cloud. Then the cloud performs fine grained encryption over the encrypted data provided by the data owner based on the ACPs provided by the data owner. It should be noted that the idea of two layer encryption is not new. However, the way we perform coarse and fine grained encryption is novel and provides a better solution than existing solutions based on two layers of encryption [8]. We elaborate in details on the differences As the data owner does not keep a copy of the data, whenever user dynamics changes, the data owner needs to download and decrypt the data, re-encrypt it with the new keys, and upload the encrypted data. The user dynamics refers to the operation of adding or revoking users. Notice also that this process must be applied to all the data items encrypted with the same key. This is inefficient when the data set to be re-encrypted is large.

between our approach and existing solutions in the related work section. A challenging issue in the TLE approach is how to decompose the ACPs so that fine-grained ABAC enforcement can be delegated to the cloud while at the same time the privacy of the identity attributes of the users and confidentiality of the data are assured. In order to delegate as much access control enforcement as possible to the cloud, one needs to decompose the ACPs such that the data owner manages minimum number of attribute conditions in those ACPs that assures the confidentiality of data from the cloud.Each ACP should be decomposed to two sub ACPs such that the conjunction of the two sub ACPs result in the original ACP. The two layer encryption should be performed such that the data owner first encrypts the data based on one set of sub ACPs and the cloud re-encrypts the encrypted datausing the other set of ACPs. The two encryptions together enforce the ACP as users should perform two decryptions to access the data.

Example 1: We use the following running example where ahospital (Owner) supports fine-grained access control on electronic health records (EHRs) and makes these records available to hospital employees (Usrs) through a public cloud (Cloud). Typical hospital employees includes Usrs playing different roles such as receptionist (rec), cashier (cas), doctor (doc), nurse (nur), pharmacist (pha), and system administrator (sys). An EHR document consists of data items including Billing Info (BI), Contact Info (CI), Medication-Report (MR), Physical Exam (PE), Lab Reports (LR), and Treatment Plan (TP) and so on. In accordance with regulations such as health insurance portability and accountability act (HIPAA), the hospital policies specify which users can access which data item(s). In our example system, there are four attributes, role (rec, cas, doc, nur, pha, sys), insurance plan, denoted as ip, (ACME, MedA, MedB, MedC), type (assistant, junior, senior) and year of service, denoted as yos, (integer).

The data owner enforces the former by encrypting the data for the users satisfying the former and the cloud enforces the latter by re-encrypting the data owner encrypted data for the users satisfying the latter. Since the data owner handles the minimum number of attributes, the overhead of managing Design Goals are given in Section 3. Section 4 reports experimental results for policy decomposition algorithms and the SLE vs. the TLE approaches. Finally, Section 5 concludes the paper providing future directions.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

II.A NEW APPROACH TO MANAGE GROUP ENCRYPTION KEYS

An approach to support fine-grained selective ABAC is to identify the sets of data items to which the same access control policy (or set of policies) applies and then encrypt each such set with the same encryption key. The encrypted data is then uploaded to the cloud and each user is given the keys only for the set(s) of data items that it can access according to the policies. Such approach addresses two requirements: (a) protecting data confidentiality from the cloud; (b) enforcing fine-grained access control policies with respect to the data users. A major issue in such an approach is represented by key management, as each user must be given the correct keys with respect to the access control policies that the user satisfies. One approach to such issue is to use a hybrid solution whereby the data encryption 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 keeping multiple encrypted copies of the same key; it incurs high computational costs; it requires additional attributes to support revocation.

Therefore, a different approach is required.It is also worth noting that a simplistic group key management (GKM) scheme by which the content publisher directly delivers the symmetric keys to the corresponding attributes at the data owner is reduced. The overhead is further reduced, as it has to enforce only subsets of complex policies and thus only needs to perform a coarse grained encryption to enforce the simplified policies. Since the cloud does not handle the condition "role = doc", it cannot decrypt owner encrypted data and thus confidentiality is preserved. Notice that users should satisfy the original ACP to access the data by performing two decryptions. In this paper, we show that the problem of decomposing ACPs such that the data owner manages the minimum number of attribute conditions while at the same time assuring the confidentiality of the data in the cloud is NP-complete. We propose two optimization algorithms to find the near optimal set of attribute conditions and decompose each ACP into two sub ACPs.The TLE approach has many advantages when user dynamics changes, only the outer layer of the encryption needs to be updated. Since the outer layer encryption is performed at the cloud, no data transmission is required between the data owner and the cloud.

Further, both the data owner and the cloud service utilize a broadcast key management scheme [9] whereby the actual keys do not need to be distributed to the users. Instead, users are given one or more secrets which allow them to derive the actual symmetric keys for decrypting the data. The rest of the paper is organized as follows. Section 2 describes A New Approach to Manage Group Encryption Keys. System Model and users has some major drawbacks with respect to user privacy and key management. On 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 when the number of users becomes large and multiple keys need to be distributed to multiple users. The goal of our work is to develop an approach which does not have these shortcomings. 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 issues. Based on this idea, we have defined a new GKM scheme, called broadcast GKM (BGKM), and given a secure construction of the BGKM scheme. The idea is to give 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 access control policies can be performed efficiently and only requires updating the public information. Our BGKM scheme is referred to as access control vector BGKM (ACV-BGKM).The idea of ACV-BGKM is to construct a special matrix A where each row is linearly independent and generated using each user's secret. The group controller generates the null space Y of this matrix by solving the linear system AY = 0, randomly selects a vector in the null space, and hides the group symmetric key inside this vector. We call this vector as access control vector (ACV) and is part of the public information. An authorized user can generate a vector in the row space of the special matrix using its secret and some public information. We call this vector as key extraction vector (KEV). The system is designed such that the inner product of ACV and KEV allows authorized users to derive the group symmetric key. We show that a user who does not have a valid secret has a negligible probability of generating a valid KEV and deriving the group key. When a user is revoked, the group controller simply updates the special matrix excluding the revoked user and generates a new ACV hiding a new group key. Notice that such revocation handling does not affect the existing users as only the public information is changed.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Using the ACV-BGKM scheme as a building block, we have constructed a more expressive scheme called attribute based GKM (AB-GKM). The idea is to generate an ACV-BGKM instance for each attribute and combine the instances together using an access structure that represents the attribute based access control policy. The AB-GKM scheme satisfies all the properties of the ACV-BGKM scheme and consists of the following five algorithms: Setup, Sec Gen, Key Gen, Key Der and Update.

•Setup (ℓ): It initializes the BGKM scheme using asecurity parameter ℓ. It also initializes the set of used secrets S, the secret space SS, and the key space KS.

•Sec Gen (user, attribute): It picks a random bitstring s / S uniformly at random from SS, adds s to S and outputs s. A unique secret is assigned to per user per attribute. These secrets are used by the group controller to generate the group key and also by the users to derive the group key.

•Key Gen(S, Policy):It picks a group keykuniformlyat random from the key space KS and outputs the public information tuple PI computed from the secrets in S that satisfy the policy and the group key k. PI indirectly encodes the policy such that a usercan use PI along with its secrets only if the user satisfies the policy used to generate PI.

•Key Der (s, PI): It takes the user's secretsand thepublic information PI to output the group key. The derived group key is equal to k if and only if s S and satisfies the policy.

•Update(S): Whenever the set S change, a new groupkey k′ is generated. Depending on the construction, it either executes the Key Gen algorithm again or incrementally updates the output of the last Key Gen algorithm.

III.SYSTEM MODEL AND DESIGN GOALS:

A.System Model:

The system model consists of three different entities: as illustrated in below Fig.2, the cloud, Owner (i.e., the company manager), and a large number of Users (i.e., the staffs) . Cloud is operated by CSPs and provides priced abundant storage services. However, the cloud is not fully trusted by users since the CSPs are very likely to be outside of the cloud users' trusted domain.

Owner takes charge of system parameters generation, user registration, user revocation, and revealing the real identity of a dispute data Owner. Users are a set of registered People that will store their private data into the cloud server and share them with others in the Cloud.

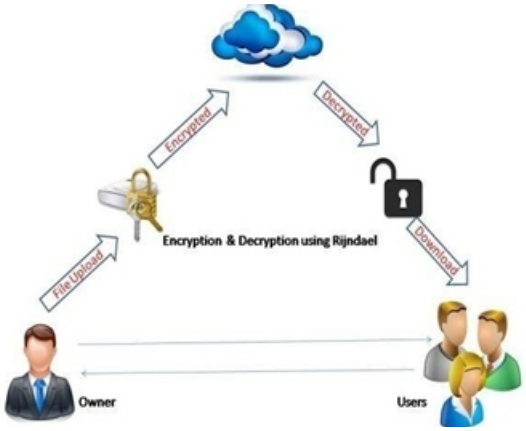


Fig.2. Encryption and decryption model.

B.Design Goals:

In this section, we describe the main design goals of theproposed scheme including Cloud Storage, Data Encryption & Decryption, Access Policy Inheritance, and Secret Key.

C.Cloud Storage:

Cloud storage is a model of networked enterprise storagewhere data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualized the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers and multiple locations. The safety of the files depends upon the hosting companies, and on the applications that leverage the cloud storage.

D.Data Encryption & Decryption:

Data confidentiality requires that unauthorized users-including the cloud are incapable of learning the content of the stored data. An important and challenging issue for data confidentiality is to maintain its availability for dynamic groups.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Specifically, new users should decrypt the data stored in the cloud before their participation, and revoked users are unable to decrypt the data moved into the cloud after the revocation. We utilized the AB-GKM scheme with the subset cover optimization. We used the complete subset algorithm

E.Access Policy Inheritance:

Access policy inheritance consists of two basic conceptualsteps: First, domain experts have to identify dependencies between incoming and outgoing attributes for each operator. We model these dependencies in a graph as given for our scenario. Second, an operator maps all access requirements specified for each of its incoming attributes to the access policy of all dependent outgoing attributes. In our example scenario, operator determines the value of the Cloud Data attribute. It needs to map the Requirement(Cloud Storage, , {Owner, User})Which is associated to the destination to the Cloud Storage attribute, since User are dependent on Cloud Storage Hence, only operators hosted by the Owners or the customer can access the attribute. After creating the new access policy for the Cloud company operator, the access policy contains the following entries.{(Cloud, (Cloud Storage, ,{customer})), (est Arrival Time, (Cloud ={Cloud Storage)}Since each operator is forward only event streams whose event attributes are annotated with consolidated access policies, it is sufficient to consider possible dependencies between events attributes of incoming streams and outgoing streams.

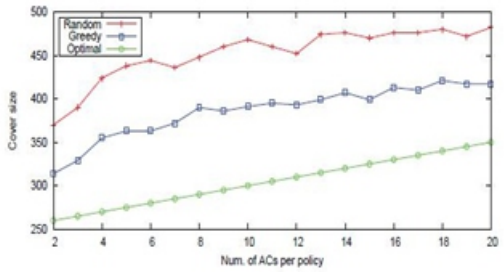
F. Secret Key:

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption.

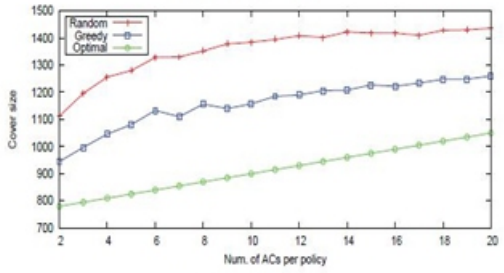
IV. EXPECTED RESULTS:

In this section we first present experimental results concerning the policy decomposition algorithms.

We then present an experimental comparison between the SLE and TLE approaches. The experiments were performed on a machine running GNU/Linux kernel version 2.6.32 with an Intel Core TM 2 Duo CPU T9300 2.50GHz and 4 Gbytes memory. Only one processor was used for computation. Our prototype system is implemented in C/C++. We use V.Shoup's NTL library version 5.4.2 for finite field arithmetic, and SHA-1 and AES- 256 implementations of Open SSL version 1.0.0d for cryptographic hashing and incremental encryption. We use boolstuff library version 0.1.13 to convert policies into DNF. Adjacency list representation is used to construct policy graphs used in the two approximation algorithms for finding a near optimal attribute condition cover. introduced by Naor et al. [11] as the subset cover we assumed that 5% of attribute credentials are revoked for the AB-GKM related experiments. All finite field arithmetic operations in our scheme are performed in a 512-bit prime field. We set the total attribute condition count between 100-1500 and the attribute conditions per policy count between 2-20. We generate random Boolean expressions consisting of conjunctions and disjunctions as policies. Each term in the Boolean expression represents an attribute condition.



(a) 500 attributes



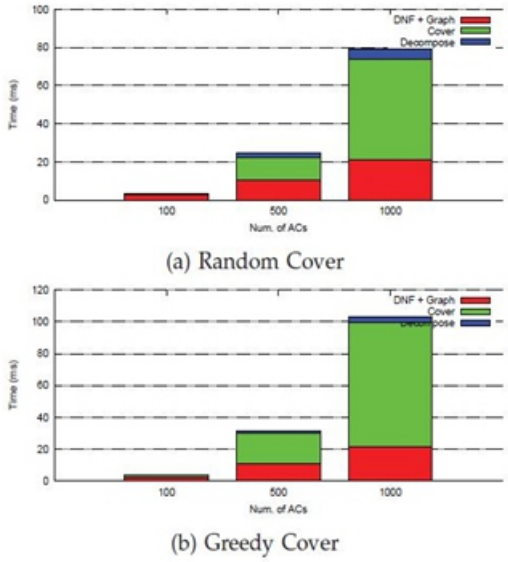
(b) 1500 attributes

Fig.3. Size of ACCs for different number of ACs.

Fig.3 shows the size of the attribute condition cover, that is, the number of attribute conditions the data owner enforces, for systems having 500 and 1500 attribute conditions as the number of attribute conditions per policy is increased. In all experiments, the greedy policy cover algorithm performs better that the random cover algorithm.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

As the number of attribute conditions per policy increases, the size of the attribute condition cover also increases. This is due to the fact that as the number of attribute conditions per policy increases, the number of distinct disjunctive terms in the DNF increases.Fig.4 shows the breakdown of the running time for the complete policy decomposition process. In this experiment, the number of attribute condition is set to {100, 500, 1000} and the maximum number of attribute conditions per policy is set to 5. The total execution time is divided into the execution times of three different components of our scheme.Fig.5 reports the average time spent to execute the AB-GKM: Key Gen with SLE and TLE approaches for different group sizes. We set the number of attribute conditions to The "DNF + Graph" time refers to the time required to convert the policies to DNF and construct a in-memory graph of policies using an adjacency list. The "Cover" time refers to the time required to to find the optimal cover and the"Decompose" time refers to time required to to create the updated policies for the data owner and the cloud based on the cover. As can be seen from the graphs, most of the time is spent on finding a near optimal attribute condition cover. It should be noted that the random approximation algorithm runs faster than the greedy algorithm. One reason for this behavior is that each time the latter algorithm selects a vertex it iterates through all the unvisited vertices in the policy graph, whereas the former algorithm simply picks a pair of unvisited vertices at random. Consistent with the worst-cast running times, the "DNF + Graph" and "Decompose" components demonstrate near linear running time, and 'the 'Cover" component shows a non-linear running time.



(a) Random Cover

(b) Greedy Cover

Fig.4. Time break down for decomposing policies.

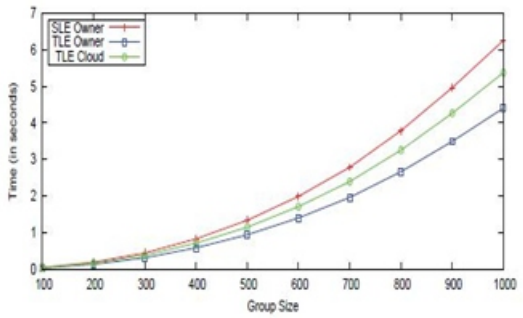


Fig.5. Average time to generate keys for the two Approaches.

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 Fig.6, a similar pattern is observed in the AB-GKM: Key Der as well.

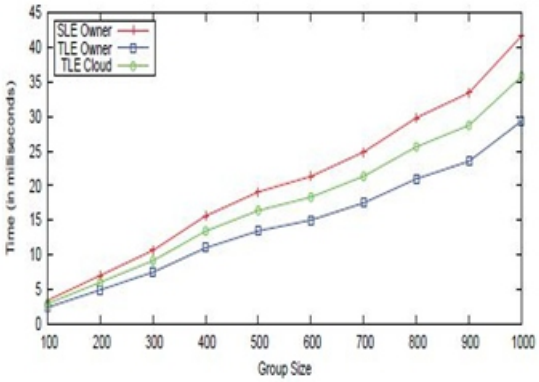


Fig.6. Average time to derive keys for the two approaches.

V. CONCLUSION:

Current 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 change.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 USRS 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 a 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 attribute based ACPs. As the experimental results show, decomposing the ACPs and utilizing the two layer of encryption reduce the overhead at the Owner. 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.

VI. REFERENCES:

[1]Mohamed Nabeel, Elisa Bertino Fellow, IEEE, "PrivacyPreserving Delegated Access Control in Public Clouds",IEEE Transactions on Knowledge and Data Engineering.

[2]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.

[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]G. Miklau and D. Suciu, "Controlling access to published data using cryptography," in VLDB '2003: Proceedings of the 29th international conference on Very large data bases. VLDB Endowment, 2003, pp. 898–909.

[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 Work sharing, ser. Collaborate Com '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]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.

[9]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.

[10]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.

[11]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.

[12]J. Li and N. Li, "OACerts: Oblivious attribute certificates," IEEE Transactions on Dependable and Secure Computing, vol. 3, no. 4, pp. 340–352, 2006.