ISSN 2278-3091

**International Journal of Advanced Trends in Computer Science and Engineering**, Vol.3 , No.5, Pages : 414- 418 (2014)
*Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh*

# A Double Layer Security Scheme for Conserving Private Data in Cloud Services

**[1]M.Santhoshi, [2]V.Srikanth**
[1]Final M.Tech Student, [2]Associate Professor
[1,2]Dept of Computer Science and Engineering
[1,2]Pydah College of Engineering, Visakhapatnam, AP, India

**Abstract:** In cloud services data security is a major problem. In these services intruders attack on keys and eventually capture the content in the network. So we introduced a novel protocol that reduces data leakage attacks. We introduced a double layer protection that used symmetric key cryptographic techniques secure data. In this process data is a encrypted twice and store in a cloud. Only the authenticated users access the information which limited the user privileges over the cloud services.

## INTRODUCTION

There is an ever-increasing amount of data available in digital form, and almost invariably that data is now near a network. Recent research of integration systems and peer-to-peer databases has created new ways for diverse groups to share and process data. But in most practical cases, complex constraints of trust and confidentiality exist between these cooperating or competing groups. As a result, in many cases data is disseminated only when there are no security or confidentiality issues among any possible recipient. This means the data that could safely be disseminated to certain parties remains hidden behind a firewall or a server.

The trust, confidentiality and security issues involved in sharing data are immense, however there are few a tools for managing access to the data that peers are encouraged or required to share (especially when distributed data processing is needed). In this paper we introduce a framework to allow a database owner to express rich access policies, generate a single, partially-encrypted version of the data that enforces all access policies, publish it, and then enforce the policies by granting keys.

A critical issue in such a context is represented by the fact that very often identity attributes encode privacy-sensitive information and this information has to be protected, even from the party distributing the contents. The privacy is considered as key requirement in all solutions and initiatives for digital identity management. It is important to point out that because of the problem of insider threats and today recognized as a major source of data theft and privacy breaches, identity attributes should still be strongly protected even if the party is distributing the contents and the content recipients belong to the same organization. The problem of disseminating contents to user groups by enforcing attribute-based access control at the same time assuring the privacy of the user identity attributes has not been addressed.

The Internet and the Web have enabled tools and systems for quickly disseminating data, by posting on Web sites or broadcasting in the user communities in a large variety of application domains and for different purposes. Whatever is because of legal requirements, organizational policies or commercial reasons and selective access to data should be enforced in order to protect data from unauthorized accesses. The modern access control models like XACML and allows one to specify access control policies that are expressed in terms of conditions concerning the protected objects against properties of subjects referred to as identity attributes and the characterizing the users accessing the protected data.

Examples of identity attributes include the role that a user has in his/her organization and the country of origin. A user thus verifies the given access control policy and identity attributes, verify the conditions of the policy. The use of such an approach is crucial to simplify access control administration and support high-level policies closer to organizational policies and is in line with current initiatives for digital identity management.

Many benefits, including on-demand provisioning that enables organizations to grow efficiently and in a cost effective manner, have been the force driving for many organizations to move into the cloud. Storage as a service (SaaS) and Data as a service (DaaS) are emerging cloud services by which organizations can seamlessly store data in the cloud and retrieve them based on access control policies (ACPs) that cover legal requirements and organizational policies. While SaaS provides a virtual storage, DaaS provides a higher level interface to store and query data on a data structure. AmazonS3 and Microsoft Azure storage services are two such popular services currently available.

While cloud data services provide many benefits, data privacy and security issues have been major concerns. Data stored in the cloud often encode sensitive information and should be protected as mandated by various organizational policies and legal regulations. A commonly adopted approach to address security and privacy is to encrypt the data before uploading them to the cloud. Encryption itself is non sufficient as often organizations have to enforce fine-grained access control on the data. Such control is often based on information such as the role of data users in the organization, projects on which users are working and so forth. Therefore, an important requirement is to support fine-grained access control, based on policies specified in expressive access control language and encrypted data hosted in the cloud.

In particular, an expressive access control model, such as XACML, allows one to specify ACPs on protected objects in terms of the properties of subjects, referred to as identity attributes. The email address, the role a user plays in an organization, age and the location a user accesses from are a few examples of such identity attributes. The identity attributes that subjects should possess in order to access protected objects are referred to as conditions. Such an attribute based access control model is crucial in order to support fine-grained access control policies to data.

## RELATED WORK

Approaches closely related to our work have been investigated in three different areas: selective publication and broadcast of documents as well as attribute-based security and group key management. The database and security communities have carried out extensive research concerning techniques for the selective dissemination of documents based on access control policies. These approaches fall in the following two categories.

1) Encryption of different subdocuments with different number of keys which are provided to users at the registration phase, and broadcast the encrypted sub documents to all users.

2) Selective multi-cast of different sub documents to different user groups and where all sub documents are encrypted with one symmetric encryption key.

The latter approaches assume that the users are honest and do not try to access the sub documents to which they do not have access authorization. Therefore, these approaches provide neither backward nor forward key secrecy. In the traditional approaches users are able to decrypt the sub documents for which they have the keys. However, such approaches require allow some keys be distributed in advance during user registration phase. This requirement leads difficult to assure forward and backward key secrecy when user groups are dynamic with frequent join and leave operations.

Further, the rekey process is not transparent therefore shifting makes burden of acquiring new keys on existing users when others leave or join. In this case our approach makes rekey transparent to users by not distributing actual keys during the registration phase. Attribute-Based Encryption (ABE) is another approach for implementing encryption based access control to documents. Under such type of approach the users are able to decrypt sub documents if they satisfy certain policies. ABE has two variations: the associating encrypted documents with attributes and user keys with policies; associating user keys with attributes and encrypted documents with policies.

In either case the cost of key management is minimized by using attributes that can be associated with users. These approaches require the attributes considered in the policies to be sent in clear. It contains clear texts that reveals sensitive information about users during both registration and document distribution phases. In contrast, our approach preserves user privacy in both phases in that users are not required to reveal the values of their identity attributes to the content distributor. Group Key Management (GKM) is a widely investigated topic in the context of group-oriented multicast applications. Early work on GKM relied on a key server to share a secret with users to distribute keys to decrypt documents. Such approaches suffer from the drawback of sending O(n) rekey information and where n is the number of users, in the event of join or leave to provide forward and backward secrecy. Hierarchical key management schemes, are the key server hierarchically establishes secure channels with different sub-groups instead of with unique users are introduced to reduce this overhead.

However, they only reduce the size of the rekey information to O (log n) and each user needs to manage at worst O (log n) hierarchically organized redundant keys. Similar to the spirit of our approach, there have been efforts to make rekey a one-off process. The secure lock approach based on the Chinese Remainder Theorem (CRT) performs a single broadcast to rekey. However, the proposed approach is inefficient for large n values as it requires performing CRT calculation involving congruence's each time a new document is sent. The access control polynomial approach encodes secrets given to users at registration phase in a special polynomial of order at least n in such a way that users can derive the secret key from this polynomial. The special polynomials used in this approach represent only a small subset of domain of all the polynomials of order n and security of the approach is neither fully analyzed nor proven.

To delegate policy changes enforcement to the server, avoiding re-encryption for the data owner, we adopt a two layer encryption approach. The owner encrypts the resources and sends them to the server in encrypted form, the server can impose another layer of encryption (following directions by the data owner).We then distinguish two layers of encryption.

Base Encryption Layer (BEL), performed the data owner before transmitting data to the server. It enforces encryption on the resources according to the policy existing at initialization time.

• Surface Encryption Layer (SEL), performed by the server over the resources already encrypted by the data owner. It leads to the dynamic changes over the policy. Both layers enforce encryption by means of a set of symmetric keys and a set of public tokens between these keys.

At BEL level we distinguish two kinds of keys: derivation keys and access keys. Access keys are the ones actually used to encrypt resources and derivation keys are used to provide the derivation capability via tokens, i.e., tokens can be defined only with the derivation key at starting point. Each derivation key k is always associated with an access key k an obtained by applying a secure hash function to k, that is, a = h (k). In other words, keys at the BEL level always go in pairs, hk, kai. The rationale for this evolution is to distinguish the two roles associated with keys, namely: enabling key derivation (applying the corresponding tokens) and enabling resource access.

In the SEL level there is no distinction between derivation and access keys (intuitively single key carries out both functions).Again, we define a key derivation function ∗s : K 7→ 2Kthat can be represented by means of a graph having a vertex for each key k defined at SEL and an edge connecting vertices(si, sj ) if there is a token in the public catalog allowing the derivation of sj .k from si.k. A key assignment is a function -s : U ∪ R 7→ K ∪ {null}that associates with each user u∈ U the (single) key released to the user by the server and with each resource r∈ R the(single) key with which the resource is encrypted by the server (if any).

III. PROPOSED WORK

In this proposed work we introduced double layer security method. We propose a new model which contains Cloud Service, user, auditor, and data owner.

Cloud Service: It is a third party service which contains large amount of database to store information. It allows data to store information by authenticated users.

User: He / She only have an access to read data when request to cloud.

Data Owner: He / She stores encrypted content and generate signature to the content. He is the owner of the stored content.

Auditor: He / She verifies the content in the cloud frequently, to check if content is secure or not. He can check file using the signature sent by cloud and Data owner.

Algorithm is as follows:

**a) Initialization**

Initially all users, Data owners register in the cloud service. For data owners the Cloud generate a secret key 'sk'. And it is modified by a random value 'r' by doing simple arithmetic operation. That modified key 'msk' is sent to data owner such as (msk,r). Data owner receives that modified key and reveal the modified key.

**b) Storage and encryption**

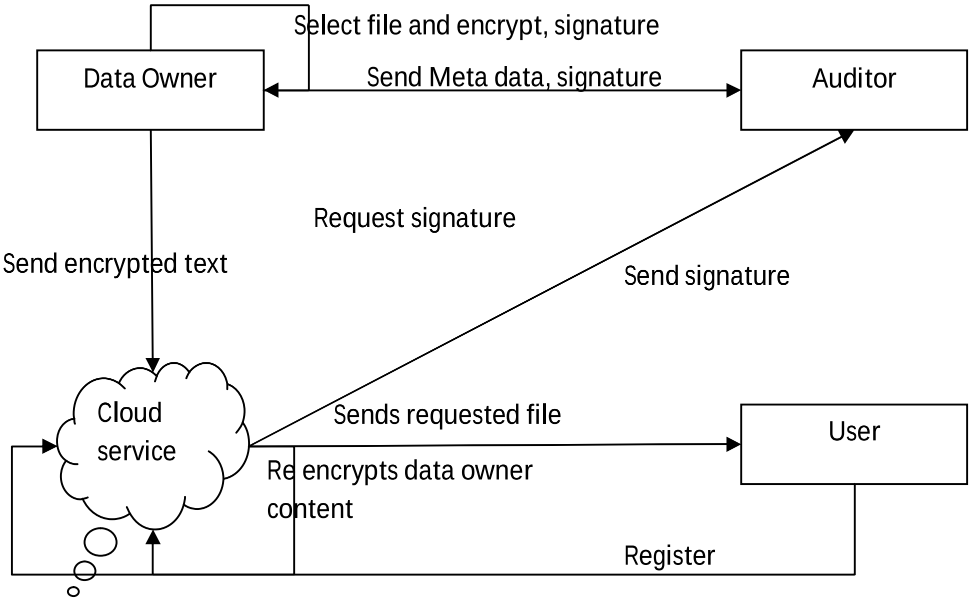
After generating of the key, Data owner selects a text file and encrypts that file content using Advanced Encryption Standard. AES operates on a 4×4 column-major order matrix of bytes, termed the state have a larger block size and have additional columns in the state. The general AES calculations are done in a special finite field.

The key size used for an AES cipher specifies the number of repetitions transformation rounds that convert the input is so called the plaintext. The final output called the cipher text. The numbers of cycles of repetition are as follows: 10 cycles for 128-bit keys per repetition,12 cycles for 192-bit keys per repetition, 14 cycles for 256-bit keys per repetition.

Each round consists of several processing steps and each containing four similar but different stages that includes one that depends on the encryption key itself. A sequence set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

**First layer encryption Algorithm:**

```
Cipher (byte in[16], byte out[16], key_arrayround_key[Nr+1])
Begin
Byte state[16];
State = in;
AddRoundKey (state, round_key [0]);
for i = 1 to Nr-1 stepsize 1 do
SubBytes (state);
Shift Rows (state);
Mix Columns (state);
AddRoundKey (state, round key[i]);
End for
Sub Bytes (state);
Shift Rows (state);
AddRoundKey (state, round_key[Nr]);
End
```



After generating cipher text Data owner generates signature by using secure hash algorithm. The signature is send to Auditor and encrypted the file content sent to cloud.

**Second layer encryption algorithm**

In this we used advanced version of AES algorithm that is Rijandeal algorithm.

**Algorithm**

```
Round (State, Expanded KEY[i])
{
Substitute Bytes (State);
Shift Rows (State);
Mix Columns (State);
Add Key (State [], Expanded KEY[i]);
}
Last Round (State, Expanded_KEY[Nr])
{
Substitute_Bytes (State);
Shift_Rows (State);
Add Key (State [], Expanded KEY[i]);
}
```

After encrypting the cipher text sent by the Data owner is encrypted by the cloud and store the final encrypted content in the cloud service.

**b)Verification**

In this process the auditor mainly involves mainly in the process of verification. Auditors have the signature sent by the data owner and request the cloud for stored file signature. In a cloud decrypts the encrypted content and generate signature and sent to auditor. Then auditor compares the signatures of data owner and cloud of requested file. If the signatures are same the files are secure, otherwise auditor conclude that file in the cloud is corrupted and sends status to data owner.

By using this process we can reduce the maximum leakage of data and man in the middle attacks in the network channel.

## CONCLUSION

In this paper we introduced double layer security method to prevent and reduce the leakage problems. In this we used symmetric key encryption techniques to achieve secure key. And the content also encrypted twice for reducing corruption of the content. We strictly allow authenticated users only to access the cloud service. By using this we can reduce man in the middle attacks. In process the computation and communication also reduced.

## 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] E. Bertino and E. Ferrari, "Secure and selective disseminationof XML documents," ACM Trans. Inf. Syst. Secur., vol. 5, no. 3,pp. 290–331, 2002.

[3] G. Miklau and D. Suciu, "Controlling access to publisheddata using cryptography," in VLDB '2003: Proceedings of the29th international conference on Very large data bases. VLDBEndowment, 2003, pp. 898–909.

[4] N. Shang, M. Nabeel, F. Paci, and E. Bettino, "A privacy preserving approach to policy-based content dissemination,"in ICDE '10: proceedings of the 2010 IEEE 26th International Conference on Data Engineering, 2010.

[5] M. Nabeel, E. Bertino, M. Kantarcioglu, and B. M. Thuraisingham,"Towards privacy preserving access control in thecloud," in Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing,ser. CollaborateCom '11, 2011, pp. 172–180.

[6] M. Nabeel, N. Shang, and E. Bertino, "Privacy preserving policybased content sharing in public clouds," IEEE Transactionson Knowledge and Data Engineering, 2012.
[7] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in Proceedings of the33rd International Conference on Very Large Data Bases, ser. VLDB'07. VLDB Endowment, 2007, pp. 123–134.
[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 ofthe 13th Annual International Cryptology Conference on Advancesin 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 the21st Annual international Cryptology Conference on Advances inCryptology, ser. CRYPTO '01. London, UK: Springer-Verlag, 2001, pp. 41–62.[11] J. Li and N. Li, "OACerts: Oblivious attribute certificates,"IEEE Transactions on Dependable and Secure Computing, vol. 3,
no. 4, pp. 340–352, 2006.
[12] T. Pedersen, "Non-interactive and information-theoretic secureverifiable secret sharing," in CRYPTO '91: Proceedings of the11th Annual International Cryptology Conference on Advances inCryptology. London, UK: Springer-Verlag, 1992, pp. 129–140.
[13] M. Nabeel and E. Bertino, "Attribute based group key management,"IEEE Transactions on Dependable and Secure Computing, 2012.
[14] A. Shamir, "How to share a secret," The Communication ofACM, vol. 22, pp. 612–613, November 1979.

## BIOGRAPHIES

V.Srikanth is currently working as Associate Professor in Pydah College of Engineering, Visakhapatnam, AP, India. He has 17 years of experience and his interested areas are cloud computing, network security.

M.Santoshi pursuing M.Tech in Pydah College of Engineering, Visakhapatnam, AP, India. Her interested areas are cloud computing, network security.