ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 4, Issue 12, December 2016

Secure Data Sharing in Clouds by Using Attribute Based Encryption

Priti Kanhere[1], Prof. Sonali Patil[2]

M.E Student, Dept. of Computer, JSPM's BSIOTR, Pune, India [1]

Asst. Professor, Department of Computer, JSPM's BSIOTR, Pune, India [2]

ABSTRACT: Most present security arrangements depend on edge security. Be that as it may, Cloud registering breaks the association edges. At the point when information dwells in the Cloud, they live outside the authoritative limits. This leads clients to a loss of control over their information and raises sensible security worries that back off the appropriation of Cloud figuring. Is the Cloud specialist co-op getting to the information? Is it honest to goodness applying the get to control strategy characterized by the client? This paper exhibits an information driven get to control arrangement with enhanced part based expressiveness in which security is centred on ensuring client information in any case the Cloud specialist co-op that holds it. Novel personality based and intermediary re-encryption methods are utilized to secure the approval show. Information is scrambled and approval standards are cryptographically secured to protect client information against the specialist co-op get to or bad conduct. The approval show gives high expressiveness part progressive system and asset chain of command support. The arrangement exploits the rationale formalism gave by Semantic Web advances, which empowers propelled control administration like semantic clash identification.

KEYWORDS: Data-centric security, Cloud computing, Role-based access control, Authorization.

I. INTRODUCTION

Security is one of the main user concerns for the adoption of Cloud computing. Moving data to the Cloud usually implies relying on the Cloud Service Provider (CSP) for data protection. Although this is usually managed based on legal or Service Level Agreements (SLA), the CSP could potentially access the data or even provide it to third parties. Moreover, one should trust the CSP to legitimately apply the access control rules defined by the data owner for other users [3]. The problem becomes even more complex in Inter-cloud scenarios where data may flow from one CSP to another. Users may loss control on their data. Even the trust on the federated CSPs is outside the control of the data owner. This situation leads to rethink about data security approaches and to move to a data-centric approach where data are self-protected whenever they reside.

Encryption is the most widely used method to protect data in the Cloud. In fact, the Cloud Security Alliance security guidance recommends data to be protected at rest, in motion and in use. Encrypting data avoids undesired accesses [6]. However, it entails new issues related to access

Control management. A rule-based approach would be desirable to provide expressiveness. But this supposes a big challenge for a data-centric approach since data has no computation capabilities by itself. It is not able to enforce or compute any access control rule or policy. This raises the issue of policy decision for a self-protected data package: who should evaluate the rules upon an access request? The first choice would be to have them evaluated by the CSP, but it could potentially bypass the rules [7]. Another option would be to have rules evaluated by the data owner, but this implies that either data could not be shared or the owner should be online to take a decision for each access request.

II. Related Work

File upload
File download
File update
New group user inclusion
Departing group user

File Upload:

Whenever a need to share data among the group arises, the owner of the file sends the encryption request to the CS. The request is accompanied by the file (F) and a list (L) of users that are to be granted access to the file. L also contains the access rights for each of the users. The users may have READ-only and/or READ–WRITE access to the file. Other parameters can be also set to enforce fine-grained access control over the data. L is used to generate the ACL for the data by the CS. L is sent to the CS only if the data are to be shared with a new proposed group. If the group already exists, the encryption request will not contain L; rather, the group ID of the existing group will be sent. The CS, after receiving the encryption request for the file, generates the ACL from the list and creates a group of the users. The ACL is separately maintained for each file. The ACL contains information regarding the file such as its unique ID, size, owner ID, the list of the user IDs with whom the file is being shared, and other metadata. If the group already existed, only the ACL for the file is created. Next, the CS generates K according to the procedure defined in Section III-B and encrypts the file with an appropriate symmetric block cipher (we have used the AES for encryption purposes). The result is an encrypted file (C). Subsequently, the CS generates Ki and K_ i for every user and deletes K by secure overwriting. Secure overwriting is a concept in which the bits in the memory are constantly flipped to make sure that a memory cell never grips a charge for enough duration for it to be remembered and recovered. The Ki for each user is inserted into the ACL for later use. To protect the integrity of the file, the CS also computes the hash-based message authentication code (HMAC) signature on every encrypted file. A similar procedure for the HMAC key is adopted. However, the HMAC key is kept by the CS only. The encrypted data, the group ID (in the case of a newly generated group), and the K_ i for the owner are sent to the requesting data owner. The group ID and the K_ i for the rest of the group users are directly sent to them over a secure communication channel. The public keys of the group users can be also used to transmit the user portion of the key. We have used the public keys of the users to transmit the key portions. The user, after receiving C, uploads it to the cloud. K is deleted via secure overwriting from the CS after the encryption process. It is noteworthy that the key generation process is executed once when the group is initiated and the first file is submitted for encryption. Moreover, a newly joining member also activates the key generation but only for the new member.

File Download:

The authorized user sends a download request to the CS or downloads the encrypted file (C) from the cloud and sends the decryption request to the CS. The cloud verifies the authorization of the user through a locally maintained ACL. The decryption request is accompanied by the user portion of the key, i.e., K_ i, along with other authentication credentials. The CS computes K by applying XOR operation over K_ i and the corresponding Ki from the ACL. As each of the users correspond to a different pair of Ki and K_ i, none of the users can use other users' K_ i to masquerade identity. Subsequently, the CS proceeds with the decryption process after verifying the integrity of the file. If the correct K_ i is received by the CS, the result will be a successful decryption process; otherwise, the decryption will fail. After successful decryption, the file is sent to the requesting user through a secure communication channel that could be Secure Sockets Layer (SSL) or Internet Protocol Security (IPSec) channels. K is deleted via secure overwriting from the CS after decryption. The users are authenticated before the request processing according to standard procedures. Similar to the file upload process, the downloading of the file can be also done by the CS on behalf of the user. In the aforesaid case, the decryption request is sent to the CS. The CS, after authenticating the user, sends the download request to the cloud for the specified file. The cloud sends the encrypted file (C) to the CS. The rest of the process for the decryption is the same.

File Update:

Updating the file has a similar procedure to that of uploading the file. The difference is that, while updating, all of the activities related to the creation of the ACL and key generation are not carried out. The user, who has downloaded the file and made any changes, sends an update request to the CS. The request contains the group ID, the file ID, and K_i, along with the file to be encrypted after changes. The CS verifies that the user has the WRITE access to the file from the corresponding ACL. In the case of a valid update request, the CS computes K by XORing Ki and K_ i, encrypts the file, and performs the HMAC calculations. The encrypted file is sent to the user or uploaded to the cloud. K is deleted afterward.

New Group User Inclusion:

If a new user joins the group, the addition of the user is made on the request of the file owner. The request contains the user ID of the joining user, along with the access control parameters to be included in the ACL, and the group ID. The parameters include the IDs of the files for which the user has been granted access rights. It also includes the details indicating the READ and/or WRITE rights granted to the user. Alternatively, the date can be mentioned from which the access rights are valid for the user. This ensures the backward access control for the joining member. The CS, after receiving the joining request, updates the ACLs related to the files for which the access is granted. The key shares are generated, and the user shares are sent to the user along with the corresponding file IDs.

Departing Group User:

The CS is notified about a departing member by the group owner. The CS removes all of the records for the departing user from the ACLs of the related files. As the whole key is not possessed by the group members, the departing member (even being malicious) will be unable to decrypt any of the group data files. Even the presence of encrypted files with a malicious departing member will not affect the privacy of the data. The malicious member will be unable to construct the whole key for decryption. Therefore, the forward access control is also ensured by the SeDaSC methodology. The next section discusses how different security services are achieved by the SeDaSC methodology.

III. PROPOSED ALGORITHM

STEP 1: **PSEUDONYM GENERATION**: THE PSEUDONYM GENERATION ALGORITHM IS RUN BY EACH USER.

Input: ID
Output: Pseudonym P

Step 2: **Convergent encryption:**

KeyGenCE(M) --> K is the key generation algorithm that maps a data copy M to a convergent key K;
EncCE(K, M) --> C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a cipher-text C;
DecCE(K, C) --> M is the decryption algorithm that takes both the cipher-text C and the convergent key K as inputs and then outputs the original data copy M;

IV. PSEUDO CODE

P - PSEUDONYM
MK - MASTER KEY
SK - PRIVATE KEY
OK - OUTSOURCIG KEY
M - MESSAGE
CT - CIPHERTEXT

SETUP (P, K) $\rightarrow$ (P, MSK) (1)

DOI: 10.15680/IJIRCCE.2016. 0412078

$\text{KEYGEN}\ (P, MSK, ID_\alpha) \rightarrow SK_\alpha$ (2)
$\text{ENCRYPT}\ (P, ID_\alpha, M) \rightarrow C_\alpha$ (3)
$\text{RKGEN}\ (P, SK_\alpha, ID_\alpha, ID_\beta) \rightarrow RK_{\alpha\rightarrow\beta}$ (4)
$\text{REENCRYPT}\ (P, RK_{\alpha\rightarrow\beta}, C_\alpha) \rightarrow C_\beta$ (5)
$\text{DECRYPT}\ (P, SK_\alpha, C_\alpha) \rightarrow M$ (6)

V. SERVEY ON

1 .Ciphertext-policy attribute-based encryption [8]

Brent Waters [8] present a new methodology for realizing Ciphertext-Policy Attribute Encryption (CPABE) under concrete and noninteractive cryptographic assumptions in the standard model. This solution allow any encryptor to specify access control in terms of any access formula over the attributes in the system. In our most efficient system, ciphertext size, encryption, and decryption time scales linearly with the complexity of the access formula.

It provides a framework for directly realizing provably secure CPABE systems. In this system, the ciphertext distributes shares of a secret encryption exponents across different attributes according to the access control LSSS matrix M.

A user's private key is associated with a set S of attributes and he will be able to decrypt a ciphertext i his attributes \satisfy" the access matrix associated with the ciphertext. As in previous ABE systems, the primary challenge is to prevent users from realizing collusion attacks.

Main tool to prevent this is to randomize each key with an freshly chosen exponent t. During decryption, each share will be multiplied by a factor t in the exponent. Intuitively, this factor should \bind" the components of one user's key together so that they cannot be combined with another user's key components. During decryption, the different shares (in the exponent) that the algorithm combines are multiplied by a factor of t. ultimately these randomized shares are only useful to that one particular key.

2. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data [9]

V. Goyal[9] develop a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In this cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. Goyal [9] provides a construction for Key-Policy ABE that was very expressive in that it allowed the policies (attached to keys) to be expressed by any monotonic formula over encrypted data. The system was proved selectively secure under the Bilinear Diffie-Hellman assumption. However, they left creating expressive Ciphertext Policy ABE schemes as an open problem.

3. Abac and rbac: Scalable, flexible, and auditable access management [10]

As user populations of information systems have expanded, the challenge of control ling access to resources using security policies has grown. Researchers and system developers have simplified the administrative process by using groups of users who have the same authorizations. User groups were the precursor to role-based access control. RBAC groups permissions into roles and requires all access to occur through the RBAC system. Groups of permissions can then be readily provided to users in the simple operation of assigning roles. An enterprise’s roles must be engineered to support security and business rules. Over time, enterprises recognized a need for going beyond RBAC’s groups of users and permissions. They needed to include attributes, such as time of day and user location, for distributed, dynamically changing systems. During this period, attribute-based access control was identified as a replacement for or adjunct to RBAC.

Role-Based Access Control

With RBAC, roles can be well understood by their names, and they determine the sets of permissions to be granted to users. In addition, it’s easy to audit which users have access to a given permission and what permissions have been granted to a given user. A limited number of roles can represent many users or user types, and roles can be assigned to users by non-expert personnel. However, roles must be engineered before RBAC can be used. Furthermore, RBAC must be constrained to handle dynamically changing attributes, such time of day and location. Core RBAC can’t handle such attributes.

Attribute-Based Access Control

With ABAC, there's no need to engineer roles as long as role names aren't used as attributes. Dynamically changing attributes, such as time of day and location, can be accommodated in access control decisions. However, a potentially large number of attributes must be understood and managed, and attributes must be selected by expert personnel. Furthermore, attributes have no meaning until they're associated with a user, object, or relation, and it's not practical to audit which users have access to a given permission and what permissions have been granted to a given user.

VI. Conclusion and Future Work

A data-centric authorization solution has been proposed for the secure protection of data in the Cloud. SecRBAC allows managing authorization following a rule-based approach and provides enriched role-based expressiveness including role and object hierarchies. Access control computations are delegated to the CSP, being this not only unable to access the data, but also unable to release it to unauthorized parties. Advanced cryptographic techniques have been applied to protect the authorization model. A re-encryption key complements each authorization rule as cryptographic token to protect data against CSP misbehavior. The solution is independent of any PRE scheme or implementation as far as three specific features are supported. A concrete IBPRE scheme has been used in this paper in order to provide a comprehensive and feasible solution.

References

1. A. Sahai and B. Waters, 'Fuzzy Identity Based Encryption. In Advances in Cryptology – Eurocrypt' , volume 3494 of LNCS, pages 457–473. Springer, 2005
2. R. Bobba, H. Khurana, and M. Prabhakaran, 'Attribute - sets: A practically motivated enhancement to attribute -based encryption', in Proc.ESORICS, Saint Malo, France, 2009.
3. S. Yu, C. Wang, K. Ren, W. 'Lou,Achiving secure, scalable, and fine-grained data access control in cloud computing', in Proc. IEEE INFOCOM 2010, 2010, pp. 534–542
4. G. Wang, Q. Liu, and J. Wu, 'Hierarchical attribute-based encryption for fine-grained access control in cloud storage services', in ACM Conference on Computer and Communications Security , E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds. ACM, 2010, pp. 735–737
5. R. Cramer and V. Shoup, 'Design and analysis of practical public key encryption schemes secure against adaptive chosen cipher text attack', SIAM J. Compute., vol. 33, no. 1, pp. 167–226, 2004.
6. M. Green, S. Hohenberger, and B. Waters, 'Outsourcing the decryption of ABE Cipher texts', in Proc. USENIX Security Symp.,San Francisco, CA, USA, 2011, p. 34.
7. J. Lai, R. H. Deng, C. Guan, and J. Weng, 'Attribute-based encryption with verifiable outsourced decryption', IEEE Trans. Inf. Forensics Secure., vol. 8, no. 8, pp. 1343–1354, Aug. 2013.
8. B. Waters, 'Cipher text-policy attribute-based encryption: An expressive, efficient, and provably secure realization', in Public Key Cryptography - PKC 2011, 2011, vol. 6571, pp. 53– 70.
9. V. Goyal, O. Pandey, A. Sahai, and B. Waters, 'Attribute-based encryption for fine-grained access control of encrypted data', in Proceedings of the 13th ACM Conference on Computer and Communications Security, ser. CCS '06, New York, NY, USA, 2006, pp. 89–98.
10. E. Coyne and T. R. Weil, 'Abac and Rbac: Scalable, flexible, and auditable access management', IT Professional, vol. 15, no. 3, pp. 14–16, 2013.
11. Dan Boneh and Matt Franklin, 'Identity-based encryption from the Weil Pairing', SIAM Journal of Computing, 32(3):586–615, 2003.
12. Markus Jakobsson, 'On quorum controlled asymmetric proxy re-encryption', In Proceedings of Public Key Cryptography, pages 112–121, 1999.

BIOGRAPHY

Priti Kanhere is a M.E Student in the Computer Engineering Department, JSPM's BSIOTR Wagholi College, Savitribai Phule Pune University. She received Bachelor Of Engineering (BE) degree in 2015 from Solapur University, Pandharpur, MS, India. Her research interests are Cloud Computing.

Sonali Patil is a Assistant Professor in Computer Engineering Department, JSPM's BSIOTR Wagholi College, Pune, MS, India. She Pursuing her PHD from BSAU Chennai.