World Applied Sciences Journal 34 (3): 307-311, 2016
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2016.34.3.123

Time Based Automatic Data Deletion in Role and Policy Based Access Control System

[1]M. Vanitha, [2]C. Kavitha and [3]C. Karthikeyan

[1]Architect, Software Development, Altimetrik India Pvt. Ltd. Chennai, India
[2]Assistant Professor, Department of Computer Science, Thiruvalluvar Govt. Arts College, Rasipuram, India
[3]Professor, Department of Electrical and Electronics Engineering,
K.S.R College of Engineering, Tiruchengode, Namakkal, India

Abstract: Cloud storage is the leading storage model, in which the data is stored across multiple logical servers that spreads across various physical locations around the globe. Physical location of the data servers are not known to the cloud users. They are owned typically by the cloud hosting company. Companies started moving to cloud to minimize the data management cost. Especially small and medium sized enterprises with limited budgets where owning data takes a toll on their budget would like to move towards cloud storage providers. In order to ensure data safety due to natural disaster and attackers, multiple copies of data are maintained in different geographical locations. Privacy and integrity of outsourced data is one of the key concerns in the recent years. Each cloud provider has diverse group of users with diverse security requirements. Restricting users from unauthorized access, providing fine grained role based access, providing data retention proof when customer do not need the data any more becomes key challenges. This has slightly slowed the cloud adoption rate. In order to address these critical challenges, we formalize a Role and Policy Based cloud access for secured and fine grained access and then propose a Time Based Automatic Data Deletion for data retention assurance.

Key words: ABE (Attribute Based Encryption) · Cloud Computing · Integrity · RBAC · Key Management · Secret Keys · Access Control

INTRODUCTION

Modern storage systems provide lot of powerful and promising encryption mechanisms to store data. But they do not always provide mechanism to reliably delete the encrypted data. In first part of this paper, we are going to discuss about the Role and Policy based access using Advanced Encryption Standard to store the data. Advanced Encryption Standard is one of the promising encryption standards used by leading cloud providers in the industry. On top of ABE, we are going to define role and policy based access to the encrypted data for fine grained access control. Role based access is one of the proven model in large-scale enterprises. There are various RBAC(Role Based Access Control) models exists, however NIST (National Institute of Standard and technology) has defined a standard RBAC Model which other researchers can further extend it based on industry requirement. Among the four models (HRBAC, CRBAC, SRBAC) defined by NIST, we are going to extend the standards defined in HRBAC to work as Hierarchical Role and Policy based access HRP-BAC. In the second part we are going to discuss about time based automatic deletion of encrypted data. Cloud storage providers do not reliably delete the stored customer critical and sensitive information. Deleting just removes the pointer from the stored data, but the data can be accessible for much longer time until it gets overwritten by next data set. So cloud users started thinking about taking control over their stored data and wanted to shred the data by themselves when they don't need it any more. The Time Based Automatic data deletion approach specified in this paper will help users to selectively erase the data associated to specific Role and Policy under the same Hierarchy.

Corresponding Author: M. Vanitha, Architect, Software Development, Altimetrik India Pvt. Ltd. Chennai, India.

Related Work: Many systems have been proposed which essentially provide various access control models that ensures secured and fine grained access to cloud data. Some methods are relatively strong to our proposed work. But delegation of keys across various users in the same organization is not efficiently handled. Secret or control keys to get the data key are accessed based on the attributes specified by the individual users. This makes key management a tedious job. Also access to data files are controlled via predefined policies.

Yang Tang, Patrick P.C. Lee *et al*. [1] Introduces policy based File Assured Deletion for cloud storage called FADE. It follows the similar approach of Attribute Based Encryption (ABE), but ABE concentrates more on storing and retrieving data based on attributes, whereas FADE concentrates on deleting data based on policy. The difference between our work and FADE is that FADE deletes the stored data by revoking the policies based on the customer request. But the proposed work in this paper concentrates on both Role and Policy based access and also deleting files based on the time defined as one of the policy parameters. So with this approach we can ensure that data will be made irrecoverable in a defined time period, may be after a few days or months from the contract end date between the customer and the provider as defined as one of the attributes in policy.

Christian Cachin, Kristiyan Haralambiev *et al*. [2] introduces Policy-based Secure Deletion which provides encryption based secure deletion for all kind of storage systems, regardless of the physical storage system. In this work, data is grouped under series of protection classes and attributes controls the access of the data under protection classes are accessed via attributes. It provides operations for protecting, accessing and securely deleting the data. Secure deletion operation is applied to the set of attributes ensuring the protection classes associated with these attributes becomes inaccessible. The difference between the work defined in [2] and the our proposed work is that, this work concentrates more on time based automatic revocation of Policy and Role based fine grained storage system.

Our Contribution: In this paper we first propose Role and Policy based access control system that follows Advanced Encryption Standards (AES) as the encryption mechanism applied on top of set of attributes. Ravi Sandhu, David Ferraiolo *et al*. [3] proposes AES as the most secured algorithm to be used (per NIST standards) which provides greater security than DES (Data Encryption Standards). We propose PKI (Public Key Infrastructure) as the mechanism for distributing the public key to the users. In the first part of the paper we define mechanism to store and access data in cloud and in the second part of the paper we propose the time based automatic deletion of the stored data.

Motivating Scenario: Company A may decide to outsource its business critical sales data to a leading cloud storage provider (Amazon), the outsourced data has to be accessed only by the employees of Company A and its peer Companies. Company may have employees of different Roles like Marketing Executives, Sales Engineer, Accounting Manager, Solution Architects, IT Managers, Technical Head and Software Engineers etc. Employees of different roles may have different permission levels based on their roles. Roles and their permissions will be defined by the organization and Users will be assigned with the roles whenever they request to gain access to the application. Once after the application access is gained, further access to data is controlled by both Role and Predefined Policies. As defined in NlST, Roles can be Flat and Hierarchical [4]. Hierarchical RBAC is the widely accepted access control mechanism whereby senior employees in the hierarchy owns bigger permission set when compared to the junior employees [5, 6]. So in our motivating scenario we propose HR—BAC. A simple example of Role and Policy Based access could be a user with Role Marketing Manager can gain access to details about particular product during his entire tenure whereas a user with Role Marketing Engineer can access to the same product only the time he works on the sales of the particular product. Hence the access here is controlled by both Roles and Policy defined by the attribute Date. Christian *et al*., [2] and Yang Tang *et al*. [1] defines deletion based on policy and attributes, whereas in our motivating scenario we are proposing deletion based on time and policy. A simple scenario for Time based automatic deletion could be, revoking access on the pricing and discounting details of a particular product from all field engineers after the Sale is over, is by defining a general policy with attribute as time (End Date) for all the users under a specific hierarchy.

TBD-RBAC Model

Notations And Conventions: The users under the organization are grouped under a hierarchy. They are denoted by Roles. For example role of the person on top level (Level 1)of the hierarchy could be Director and

following him would be the Group Managers (GM) (Level 2) reporting to Directors. The next level to GMs could be Individual managers (MR) reporting to GMs (Level 3). Level 4 would be the Tech Lead team reporting to MRS. Level 5 would be the individual Field Engineers (FE) reporting to TLs. Number of levels in an organization may further extend based on domain and employee's strength. The levels in the hierarchy are denoted by HL = {l_1, l_2, l_3, l_4.....ln}. The individual user group of the organization is denoted by U = {u_1, u_2, u_3... u_n}. Roles of the organization are denoted by RL = {r_1, r_2, r_3 … r_n}. Policies are denoted by P = {p_1, p_2, p_3...p_n}. User Files are denoted by F = {f_1, f_2, f_3... f_n} and each user file is associated with a Cipher Key CK = {k_1, k_2, k_3... k_n}. Figure 1 denotes a hierarchical Role Based graph for a sample enterprise application to understand the notations clearly.

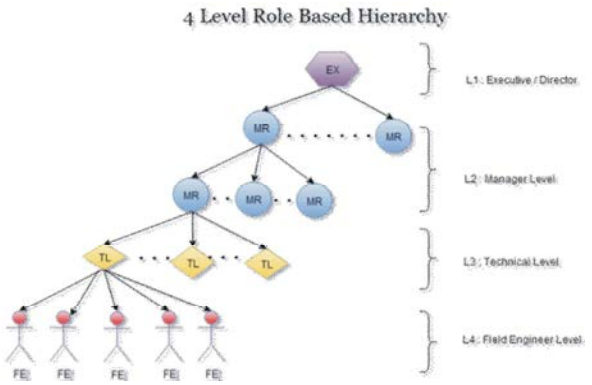

Fig. 1: Hierarchical Role Based graph

AES (Advanced Encryption Standards) Based Encryption: The TBD—RPBAC Model provides operations for protecting the files by Advanced Encryption Standards (AES) [7]. Each file is protected by a Cipher Key (K) which is stored in the Erasable Key Store of the Client Users. AES follows a symmetric key algorithm wherein the same Cipher Key is used for both Cipher as well as Inverse Cipher operations. In this work AES is proposed because of shared secret key between storage and retrieval. This makes the key management easier in contrast to public-key cryptography where in a public key is used to encrypt while storing and a private key is used to decrypt while retrieving.

Role and Policy Based Access: The goal of Role Policy Based access (RPBAC) is to control the access of resources by unidentified or unauthorized user groups. Role Based Access Control is both old and a new technology with different implementation models based on the requirement [8]. In the proposed work it is achieved by maintaining an Access Control List. Access Control List has the list of attributes associated with the files or objects. Example of such attributes could be File / Object Name, Created Date, Expiration Date, Read Access (Y/N), Write Access (Y/N), Role Based Permission etc. On top of Access Control List, the Access Policies are defined. Access Control Policies are nothing but a set of predefined rules by which users or employees of an organization are authenticated and then access to objects are either granted or denied. Files are protected by Cipher Keys. These Cipher Keys are accessed based on the predefined policies in order to access the files. For example, in an enterprise sales application, if a latest discount matrix of a leading product has to be accessed, first the policy associated with the File is accessed. Figure 2 denotes the Role based Policy graph for fine grained access control to the encrypted files.

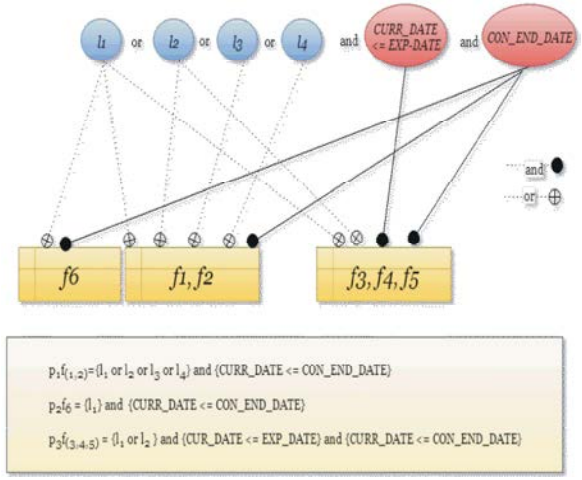

Fig. 2: Role Based Policy Graph.

Policy P_i for File F_i is denoted as p_nf_n is defined as follows

p_1f (1, 2) = {l_1 or l_2 or l_3 or l_4} and {CURR_DATE <= CON_END_DATE}

In order to access the f1, first cipher key ki has to access. ki can be accessed when rules defined in the policy file are satisfied. As per rule, users trying to access the file should be under level l1 or l2 or l3 or l4 and contract date should be still valid. If either one of this is satisfied Cipher Key can be accessed and there by decryption operation can be executed and files can be retrieved.

Time Based Automatic Deletion Schemes: Time Based Deletion provides automatic execution of deletion operation to the private key based on a predefined time-stamp and other attributes in the policy file. Deleting the files is not directly deleting the file from physical memory because it is not practically possible in cloud storage; instead it is making part of private key irrecoverable by secure diffusion and confusion mechanisms executed by the client [9]. Managing Keys separately from that of the file system is very efficient in our proposed approach, because we deal with making the key invalid instead of directly erasing the data. The Key is maintained in the client database whereas the files are stored in the cloud database. For example, if a particular discount is applicable only during a Sale period, the discount file is added with an additional expiration rule in the policy. So the policy pnfn can we re-written as

pnfn = {{l_1} OR {{l_3 or l_4 or l_5} AND {CUR_DATE BTW (START_DATE, END_DATE)}}} AND {CUR_DATE < EXP_DATE)

Where l_1 denotes the director role, l_2, l_3, l_4 denotes the GM, MR, TL roles respectively and CUR_DATE and EXP_DATE denotes Current date and Expiration date respectively.

TBD_RBAC Construction: Based on TBD—RBAC model, we define the following operations.

Create User: The algorithm Create User (attributes, ri, params) ui takes set of attributes for basic identity and authentication and if the authentication is successful and approved by an user of role ri+1 (one level up in the hierarchy), an user ui is created successfully.

Encrypt and Store File: The algorithm Encrypt (attributes, params (Key, Value), {li | 1 <= i <=N), EXP_DATE, fi, Random {K}} {efi pi} takes set of authenticating attributes for basic authentication, list of user roles li and EXP_DATE to create policy pi and list of parameters as key value pairs if additional attributes are required to define policy pi and expiration date for the file fi and a randomly generated Cipher Key. Encrypt() algorithm first creates a policy file pi based on supplied roles and attributes and then encrypts the files based on AES algorithm(Advanced Encryption Standards) using the Cipher Key K. Finally it associated the policy pi with encrypted file efi and stores efi.

Decrypt and Retrieve File: The algorithm Decrypt (pi, efi, Ki} fi takes the policy pi, encrypted file efi and the cipher key K as input. It then accesses the policy pi to check for list of rules such as roles, exp date and other attributes defined as name, value pairs and if the rule execution passes, the file is decrypted with K and retrieves original file fi. Suppose if the operation fails while executing the expiration rules defined in the policy file, the Delete () operation is invoked.

Automatic Deletion: The algorithm Delete {pi, Ki} Ki+1 takes policy file pi and the Cipher Key K as input parameters. It executes the expiration rule defined in pi and if file is past its expiration date it returns a new cipher K by again applying AES on top of ki by generating a random Cipher K, Kj to produce a new Cipher Key ki +1 which overwrites ki and thereby leaves file fi inaccessible.

Automatic Deletion operation can be executed for a single Cipher K deletion or it can be mass delete operations to make the files of an enterprise inaccessible from cloud especially when contract date ends. In these cases the contract end date can be defined as a master policy that gets executed every day invariable of the file operations and when current date meets the contract end date, based on the confirmation from a person of Role CEO and above, automatic overwriting of Cipher Keys will be executed and there by the complete set of data files become inaccessible by anyone including the cloud provider.

CONCLUSION

This paper introduces a novel security mechanism for mission critical data stored in cloud by enterprise applications. It uses policies defined based on set of roles and dynamic attributes that are specified as key- value pairs to automatically make the critical data inaccessible when organization decides to do so. This method also provides high performance fine grained access control mechanism to files. The encryption mechanism used in our work (AES) is also one of the highly effective and recommended encryption standards that is proved to be 99% cyber-attack free.

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my supervisor Dr. C. Kavitha, who gave me all her support and encouragement to do a lot of research in

cloud storage. The valuable suggestions based on multiple periodic reviews helped to improve the quality of this paper.

REFERENCES

1. Yang Tang, Patrick P.C. Lee, John C.S. Lui and Radia Perlman, 2012. FADE: Secure Overlay Cloud Storage with File Assured Deletion, in IEEE Transactions on Dependable and Secure Computing, 9(6): 901-916.
2. Christian Cachin, Kristiyan Haralambiev, Hsu-Chun Hsiao and Alessandro Sorniotti, 2013.Policy-based Secure Deletion, in Proceedings of the 2013 ACM SIGSAC conference on Computer & Communication security (CCS ' 13), pp: 259-270.
3. Elaine Barker and Quynh Dang, 2015. Recommendation for Key Management, Computer Security Division Information Technology Laboratory, National Institute of Standards and Technology (NIST), http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-57p1r3.pdf, pp: 41-71.
4. Ravi Sandhu, David Ferraiolo and Richard Khun, 2000. The NIST Model for Role-Based Access Control: Towards A unified Standard, Information Technology Lab, National institute of Standards and Technology (NlST), http://csrc.nist.gov/rbac/sandhu-ferraiolo-kuhn-00.pdf, pp: 1-12.
5. Hua Denga, Qianhong Wub, Bo Qinc, Josep Domingo-Ferrerd, Lei Zhange, Jianwei Liub and Wenchang Shic, 2014. Ciphertext-Policy Hierarchical Attribute-Based Encryption with Short Ciphertexts: Efficiently Sharing Data among Large Organizations, Information Sciences, pp: 370-384
6. Guojun Wang, Qin Liu and Jie Wu, 2010. Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services, in Proceedings of the 17th ACM Conference on Computer and Communications Security, pp: 735-737.
7. Federal Information Processing Standards Publication 197, 2001. Advanced Encryption Standard (AES), National institute of Standards and Technology (NlST), http://csrc.nist.gov/publications/fips/fips197/fips-197.gdf, pp: 13-26.
8. Stokes, E. and B. Biakley, 2000. Access Control Requirements for LDAP, http://www.rfc-base.org/txt/rfc- 2820.txt, pp: 1-8.
9. David Mazieres, Michael Kaminsky and M. Frans Kaashoek, 1999. Separating key management from file system security, in Proceedings of the seventeenth ACM symposium on Operating Systems Principles, pp: 124-139.