Ensuring Data Privacy and Access Anonymity using Cryptographic Techniques in Cloud Computing

Lamminthang Singsit
Department of Computer Science School of
Engineering and Technology, Pondicherry University,
Pondicherry, INDIA

Marie Stanislas Ashok
Systems Manager and Head
Pondicherry University, INDIA

Abstract
the main idea behind Cloud computing is to store data and other resources remotely in the cloud and accessed it anytime from anywhere through devices such as mobile devices, laptops and personal computers and any other thin client devices. This offers many advantages, like data ubiquity, flexibility of access, better performance and low start up cost. This in many ways improves the security as the cloud service provider is able to invest more in enforcing up-to date security technologies and practices than the data owner. However, this introduces new challenges with respect to ensuring the data security and the privacy. The main threat or disadvantage is that the data owner solely depends on the trustworthiness of the service provider who may not always be trustworthy. Thus, there is the need to ensure that customers data privacy and integrity are maintained. Secure access control policies, data integrity check and data privacy techniques to hide the data from the service provider needs to be implemented. Simply encrypting the data is inefficient and is vulnerable to attacks when the access control policies change. Several techniques have been proposed to address these privacy issues. Approaches based on two layers of encryption alleviate the privacy concern the disadvantage of this technique is that it still require re-encryption of the data when there is a change in the access policies. This paper presents a novel and efficient solution that employs two layers of encryption of the data and an encrypted data object containing the second access key. Changes to the access control policies are handled by re-encrypting the object containing the affected key, which is an efficient operation compared to re-encrypting the whole data or information.

Keywords:
Cloud computing, Outsourced Data, Data Security and Privacy, , confidentiality, Access Control.

1. introduction

Data owners or Organizations outsource their data to third-party service providers to reduce data management and maintenance costs. At the same time they want that the data owner or organizations must be able to have control over the data as to who should be allowed or denied access to the data. The service providers should merely take care of the management aspects, grant accesses only to the authorized users as determined by the data owner. However, data outsourcing introduces concerns for confidentiality [1]. Data should be protected from unauthorized users as well as the service provider (SP) who may be an honest but a curious operator [2]. Data must be kept secret from any unauthorized third party (Samarati and Vimercati2010).

The cloud computing model has three functional components as listed below:

1. Cloud service provider: A cloud service provider manages Cloud Storage Server(CSS), has significant storage space to store the clients' data and high computation power.
2. Client/owner: they may be individuals or organizations, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation.
3. User: It is a unit, which is registered with the owner and uses the data of owner stored on the cloud. The user can be an owner itself as well.

The data owner outsources data to the third party service provider who has the expertise and manages the data according to the policies of the data owner. The data owner grants access rights to subsets of the data objects to users who retrieve data objects from the Service Provider. Any modifications to an object made by the users with appropriate permissions are returned to the Service Provider where it replaces the current copy. The Service Provider enforces security policies to ensure that the data objects are visible only to the appropriate groups of users as determined by data owner. One important characteristic of cloud is that Membership in these groups is dynamic so users can join and leave the groups anytime. There are two essential problems that need to be addressed. First, the data should be strictly protected from unauthorized access, even from the Service Provider. Secondly, the integrity of the outsourced data must be ensured to prove that the data stored remotely is not modified or corrupted.

Cryptographic access control (CAC) has been proposed to solve the access security problems of outsourced data [2]. In this technique, the data owner encrypts the data before transmitting it to the Cloud Service Provider. This

Manuscript received May 5, 2014
Manuscript revised May 20, 2014

key is then distributed to all the users who require accesses to the data but not to the Service provider. The service Provider categorized data hierarchically by imposing a second layer of encryption to facilitate data management. This hierarchical categorization is guided by a hierarchical cryptographic key management (CKM) scheme. The second key is transmitted to the users requiring access to the data, according to the rules of access defined by the data owner. Thus, each user holds two keys, one for authentication which is also the encryption key at the server end and the other is the decryption key that is used to decrypt the data into a readable format on the user's end. This technique is not efficient as it requires the re-encryption of the data at the service Provider's end whenever there is change in group membership. This procedure is expensive, particularly when large volumes of data are involved and/or when there is a high dynamicity in the group membership. Therefore, a Cryptographic Access Control (CAC) scheme that beats the cost of re-encrypting the data when key updates occur is a desirable solution to the problem of securing outsourced data in Cloud Computing.

This paper presents a novel and efficient approach to secure the data which is described in section 3.1

2. Cloud security Requirements:

Some of the cloud security requirements are :

1. Data is encrypted and can be accessed only by the authorized users
2. Integrity of data can be checked remotely.
3. Privacy: The data should never be decrypted at the server. Decryption should be done at the user end. This will ensure that the data can be read by the authorized users.
4. Access control parameters are hidden from the service providers and other users. The service provider should only be responsible for executing the policies of the data owner. Service provider should not be able to gain access or metadata about the data when authorized accesses are performed (access anonymity).
5. The service provider does not know the number or identity of users who are allowed to access the data.
6. Communication overhead between client and server should be minimized. It should not be Computation intensive.
7. Key management should be simple.
8. It should support dynamic data operations like updation, modification, deletion, append etc as cloud is highly dynamic in nature.

The focus of this paper is to address the two main security requirements of privacy and integrity of the outsourced data.

3. Proposed model

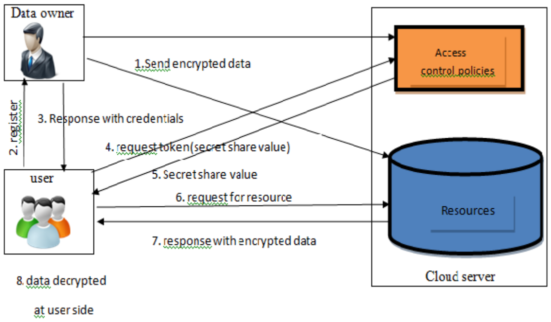

Fig 1. Basic Architecture of Preserving Data Privacy in the Cloud

3.1 A novel approach to Data Privacy and Integrity

This paper propose a novel and efficient approach to securing privacy and integrity of outsourced data by using three keys which are denoted by Bx, Kx, and Ax, where 0< x ≤ n-1 and n is the maximum number of security classes in the access control hierarchy. The first key, Bx is created by the data owner and is used to encrypted data at the owner side. This satisfies the requirement for data privacy from the unauthorized parties, including the Service Provider. In order to categorize the data received, the Service Provider creates a second key, Kx with which he encrypts the data received from the data owners. This also adds another layer of security from the unauthorized users. The third key, Ax is generated by the Service Provider and is used to encrypt the key, Kx for security reasons. The key Ax is then transmitted to the data owner who in turn distributes the key to the authorized users. Therefore each data owner and/or authorized user holds two keys, Ax and Bx. One that is used to encrypt the data before it is transmitted to the Service Provider, Bx; and the other that is used for proving authenticity to the Service Provider that, Ax. Here, the data owner is responsible for all the access policies and the service provider merely executes the policies set up by the data owner. This meets the security requirements for environment like the cloud. Another important advantage of this model is that it provides access anonymity. The service provider does not know the identity of the user who is accessing the data.

3.1(a). data confidentiality

For ensuring the confidentiality of the outsourced data, several techniques have been proposed in literature such as Fully homomorphic encryption, Hardware-anchored security, Key translation in the browser, Attribute based encryption, CryptDB and Two layer Encryption technique. This paper proposes to use the two layer encryption technique. In the first layer, the data owner performs encryption of data before outsourcing. In the second layer, another encryption is done by the service provider as part of the access control enforcement mechanism. Figure 2 shows the encryption process using symmetric key Bx (for e.g. AES)at the data owner side where data is encrypted before outsourcing to the cloud. The encrypted data is encrypted again by the cloud service provider using another symmetric key Kx. Thus, this double encryption provides a strong security against attacks than a single layer based encryption technique. The attacker needs to obtain both the secret keys to decrypt the information. Double encryption of data slows down the performance of the system but the advance in efficient encryption techniques has made the double encryption practical. Figure 2 shows the different encryption operation that a file undergoes from the data owner to the cloud service provider. At the same time, the MAC of the data is computed for checking integrity, the access control key is also generated and forwarded to the data owner.

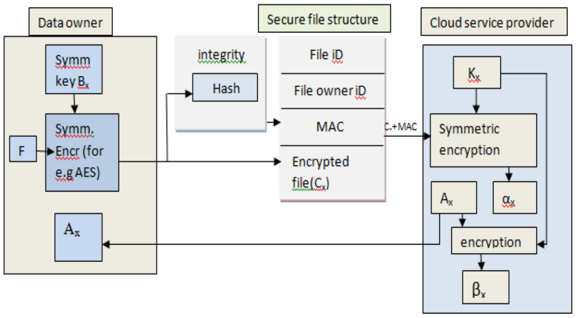

Fig 2.Preserving the privacy and integrity: processes at the owner and service provider side

3.1(b) Integrity check:

After encrypting the data, the data owner generates a message authentication code (MAC) which it transmits along with the encrypted data to cloud storage. MAC is a small fixed size block of data that is generated based on message or file of variable length using any secret key. Even a change in single character of the file leads to completely different value of MAC. The data owner uses the same key for which he encrypts the data. It is called cryptographic checksum and is used to check whether data has been tampered throughout the transmission or storage and this check can be made by the user or owner of data on retrieving the file. Fig. (2) shows the individual working of generating MAC of file F using key Bx and giving the encrypted data and its MAC on the other side. The MAC is transmitted along with the encrypted data so that users can verify that the data has not been tampered. When an authorized user receives the encrypted data, the receiver generates the MAC of received data and compares it with the MAC received along with the received data which was generated by the owner and if both the MAC codes are same then user is assured of the integrity of data. Here the verification process of data integrity can be performed as shown in Fig. (3).

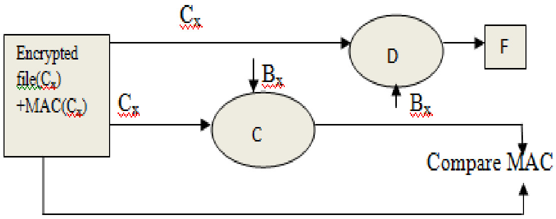

Figure 3. integrity check using MAC

Users generate the MAC of the received encrypted data using the same key Bx with which the data owner also generates the MAC. This key is shared by the data owner during the authorization of the users. The result of the MACs being same ensures that throughout the whole working, the model has taken all the required precautions and measures to protect the data from possible attacks such as data leakage, unauthorized access and tampering of data etc

3.1(c). key management

The CAC scheme is based on the concept of hierarchical CKM. CAC scheme operates in two phases namely, the setup phase in which the keys are generated in ways that enforce the security policies and the update phase which determines how key updates are handled when there is a change in access policies. A brief explanation of how hierarchical CKM schemes works is given.

The Hierarchical Key Management Model background

In the hierarchical key management model (HKM), each group in the hierarchy is assigned a unique cryptographic key. Multilevel access control system can be implemented as either dependent or independent key management model. In independent key management schemes, each class is assigned a unique key and access to lower classes is only possible if a user at a higher class holds the required lower class key [7], [11]. The dependent key management schemes on the other hand use a series of interrelated keys and access to a lower class is only possible if a user belongs to the lower class or holds a key that allows him/her to mathematically derive the required lower class key [6]. The key derivation process is performed through the application of a one-way function, so that a low level key can be derived from a high level key, but the reverse is not possible.

The above schemes have both its advantage and its disadvantage . The advantage of independent key management approach is that it is less encryption intensive than the dependent key approach, since data is only re-encrypted at the security class affected by the key update. However, it requires re-distributing the updated key to all the classes requiring the new key. The dependent key management scheme avoids distributing many keys by changing each of the keys in the sub-hierarchy associated with the affected key only. Secure key distribution can be handled by a key exchange protocol like the Diffie-Hellman key exchange scheme [10]. Another approach is to use the public key cryptography where the secret key is encrypted with the public key to ensure key privacy.

Phase 1: Setting up the System

In this phase, the data owner categorized the users into one of the several groups which is partially ordered. Each group Ui is associated with a data object F and a key Bi [3], [4]. To encrypt the data before it is sent to the SP, the data owner uses the encryption function

$$\mathrm{Enc}_{Bx}\ (F_x) = C_x$$

where $0 < x \leq n - 1$ and n is the maximum number of classes in the hierarchy.

Once the data has been encrypted, the data owner transmits it to the Service Provider for management. The Service Provider categorized all the data that it receives into classes using partial order hierarchy. The partial order hierarchy is enforced by re-encrypting each data object received with a second key, Ki which further adds another layer of security. The encryption function at the Service Provider is as follows:

$\mathrm{EncKx}\ (Cx) = \alpha x$

where $0 < x \leq n - 1$ and n is the maximum number of classes in the hierarchy.

The service provider then generates another key Ai with which he encrypts the previous key Ki giving βi. The encryption function can be represented as

$\mathrm{EncAx}\ (Kx) = \beta x$

where $0 < x \leq n - 1$ and n is the maximum number of classes in the hierarchy. Thus an encrypted object βx is obtained and can only be decrypted with the correct key Ax. The Service Provider then transmits the keys Ax to the data owner who takes care of sharing the keys with the users according to the portions of data that users are authorized to access.

Each user holds two keys, say Ai; Bi and the user accesses data, say F, by submitting their key Ai to the Service Provider for authentication. The Service Provider will use the key Ai to decrypt βi and obtain Ki that will then be used to decrypt αi to obtain Ci that is then handed to the user. To read Ci, the user will use his/her key Bi and decrypt Ci to obtain a readable form of F. So users in a group needs to maintain only two keys Ai and Bi.

Phase 2: Handling Key Updates

There is a great dynamism involved in cloud computing such as the number of users of a particular data or services, the access policies etc changes frequently. Hence frequent key updates are triggered by changes in user group membership. Once a member leaves group, subsequent access to the data object should not be allowed to this user. In this case, the data owner updates the Service Provider about the changes. The cloud service provider calculates the new access token and invalidating the previous value. This in turn is sent to the data owner who updates all the authorized users about the new access key. Key updates can now be handled more efficiently by updating the key Ax and re-encrypting only the data object containing the key Kx, instead of updating Kx and re-encrypting the data that was encrypted with Kx on the SP’s end. Moreover, avoiding data re-encryptions avoids the problem of data unavailability that arises when a data owner attempts to access data while the SP is re-encrypting it with an updated key Kx. A detailed description of handling key updates is given in the security analysis in section 4.

4. security analysis

Security analysis of the proposed model against possible attacks is considered. The data is vulnerable to threats like modification, privacy of users, confidentiality and data leakage etc. we evaluate the key management security in terms of the security of the keys Ax and Bx that the users hold, the security of the encrypted data Cx and the security of the key Kx

a). security from Cloud Service Provider

Bx encrypts the data Cx before storing in the cloud. Data owner distribute Bx only to the authorized users. Since the key Bx is kept secret from the Service Provider, the Service Provider cannot read the encrypted data Cx, and therefore the data is secured. Thus the concerned for protecting data from the service provider is addressed. Collusion attack is a potential threat to the security of the proposed scheme. Security against collusion attack can be achieved by selecting a key generation scheme, that is provably secure against collusion attack [3], [12].

b). Security against malicious outsider

Now consider a case where a malicious user say Alice is able to obtain the key Bx maliciously during its transmission to the authorized users. But since Alice does not have the authentication key Ax, she cannot exploit her knowledge of Bx to retrieve Cx unless she presents the correct Ax to the Service Provider. Also consider the case where Alice obtains the authentication key maliciously but does not have the key Bx , this too is useless since Alice cannot decrypt the data even after receiving it. Thus the only chance that an intruder can successfully obtains the plaintext data is only when he/she has both the keys Ax and Bx

c). Security of Kx

Kx is the key used to encrypt the encrypted data received from the Data Owner. The key Kx is generated by the Service Provider and it is never shared. All encryptions and decryptions with the key Kx are performed only at the Service Provider side. Therefore, Kx remains protected.

d). Granting Access and Revoking User Privileges

With respect to the security of key Ax two cases are considered:

Case 1: User who left a group attempts to retrieve and decrypt Cx : Once a user leaves the group, the data owner updates the service provider about the membership changes. The service provider in turn, immediately update the key Ax to prevent unauthorized access. To retrieve the current version of the data Cx, the user will begin by submitting his/her key, Ax, to the Service Provider for authentication. The Service Provider uses the supplied Ax to decrypt βx which is not possible because Kx has been re-encrypted with an updated Ax. Therefore, even though the key Bx was not updated, the user will be unable to retrieve Cx and decrypt it to read as shown in Figure 4.

Case 2: A malicious user intercepts the Authentication key Ax: in Figure 4, the malicious user, say Alice, intercepts Ax during its transmission to the data owner or while the data owner distributes it. She therefore proves herself to the service provider that she is an authorized user. The Service Provider then retrieves Kx by decrypting βx. key Kx obtained in turn decrypts αx and retrieve Cx that is send to Alice. However, Alice does not hold the key Bx which is needed to decrypt Cx . Thus obtaining Cx is useless. Therefore, this approach to key management, gives a strong guarantee of securing outsourced data. Thus Alice needs to steal both Ax and Bx to successfully decrypt the data Cx. However, if Ax is updated fairly frequently the cost of stealing the keys should outweigh its benefits.

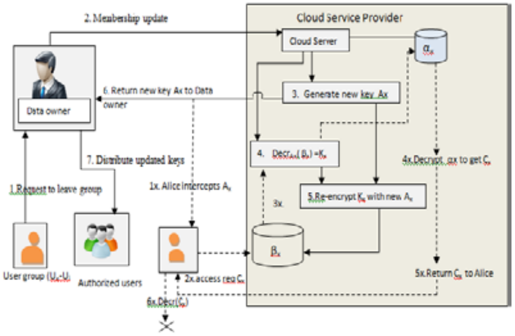

Figure 4: Handling key updates and malicious attempt to retrieve Cx

d) Tampering of data

The data is always under the threat of being tampered by any unauthorized person. To ensure the integrity of the data after transmission, MAC (Message authentication code) has been used in proposed model. The data owner generates the MAC of encrypted data before sending it and this MAC is transmitted along with the encrypted data. When receiver receives the data, the receiver can generate the MAC of received data and compare it with the MAC received along with the received data which was generated by the owner and if both the MAC codes are same then user is assured of the integrity of data, i.e., data has not been tampered.

5. Conclusion

The proposed technique provides a way for data confidentiality and privacy. A solution is also presented to the problem of handling authorization policy modifications when access control is enforced cryptographically in situations where data is stored and offered to clients by an external server. Access Anonymity and Key update problems are addressed by using three cryptographic keys, Ax;Bx; and Kx. The keys, Bx and Kx are used to encrypt data at the owner site and service provider side respectively to protect the data from both unauthorized. Service Provider is also prevented from reading the data as data owner encrypts it before storing. The key, Ax, is used to encrypt the key, Kx, which is used as a token for access anonymity. To access the outsourced data, a user presents the key, Ax to the Service Provider for authentication. Once this test is passed the user is handed the encrypted data object that can only be decrypted into a readable format if the user holds the key with which it was encrypted. So in essence, each user holds two keys, one that authenticates them to the Service Provider, Ax, and another key, Bx, that is used to read the data that they retrieve. When a situation occurs that requires a change in the authorization policy, the Service Provider reacts by creating a new key, Ax, to replace the old key, re-encrypts the key file containing Kx, and transmits the key, Ax to the data owner who in turn is responsible for distributing the new key to the remaining authorized users in the group. Since the key file is small in comparison to the data file, the cost of key updates is reduced considerably in comparison to previous approaches. An added advantage of this approach is that it removes the requirement of having to re-encrypt the data each time the authorization policy changes and therefore circumvents the problem of data consistency. Security analysis reveals that

-A user cannot access the encrypted data before he/she provides the correct secret value to the server and without revealing his or her identity

-When granting a new user to a file, the owner only needs to exchange the secret values Ax and Bx
-The method provides integrity and authenticity verifications for each file .

References

[1] S. De Capitani Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samariti. Over-encryption: Management of access control evolution on outsourced data. In In Proc. VLDB 2007, pages 123–134. Vienna, Austria, September 23-28 2007

[2] S. De Capitani Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samariti. A data outsourcing architecture combining cryptography and access control. In Proc. of the 2007 ACM Workshop on Computer Communications Security (CSAW), pages 63–69. Fairfax, Virginia, USA, November 2 2007

[3] M. J. Atallah, K. B. Frikken, and M. Blanton. Dynamic and efficient key management for access hierarchies. In Proc. ACM Conference on Computer and Communications Security, pages 190–202. Alexandria, Virginia, USA, November 7-11 2005.

[4] M. J. Atallah, M. Blanton, and K.B. Frikken. Key management for nontree access hierarchies. In Proc. of ACM Symposium on Access Control Models and Technologies, pages 11–18. Lake Tahoe, California, USA, June 7-9 2006.

[5] Jason Crampton. Cryptographically-enforced hierarchical access control with multiple keys. In Proc. of the 12th Nordic Workshop on Secure IT Systems (NordSec 2007), pages 49–60, 2007.

[6] A.V.D.M. Kayem, S.G. Akl, and P. Martin. On replacing cryptographic keys in hierarchical key management systems. Journal of Computer Security, 16(3):289–309, 2008.

[7] R.H. Hassen, A. Bouabaallah, H. Bettahar, and Y. Challal. Key management for content access control in a hierarchy. Computer Networks, 1(51):3197–3219, 2007.

[8] Nabil Giweli, Seyed Shahrestani and Hon Cheung. Enhancing Data Privacy and Access Anonymity in Cloud Computing Communications of the IBIMA Vol. 2013 (2013), Article ID 462966, 10 pages DOI: 10.5171/2013.462966

[9] Ryan, M.D., Cloud computing security: The scientific challenge, and a survey of solutions. J. Syst. Software (2013), http://dx.doi.org/10.1016/j.jss.2012.12.025

[10] Wikipedia Diffe-Hellman. Diffe-hellman key exchange. http://en.wikipedia.org/wiki/Diffie-Hellman, March 2009.

[11] W. Yu, Y. Sun, and R. Liu. Optimizing the rekeying cost for contributory group key agreement schemes. IEEE Transactions on Dependable and Secure Computing, 4(3):228–242, July-Sept.

[12] A.V.D.M. Kayem, S.G. Akl, and P. Martin. Heuristics for improving cryptographic key assignment in a hierarchy. In Proc. of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), pages 531–536. Niagara Falls, Canada, May 21-23 2007

[13] Anne V.D.M. Kayem, Patrick Martin and Selim G. Akl. Efficient Enforcement of Dynamic Cryptographic Access Control Policies for Outsourced Data

[14] Cloud Security Alliance: https://cloudsecurityalliance.org/

[15] Chunming Rong , Son T. Nguyen, Martin Gilje Jaatun, Beyond lightning: A survey on security challenges in cloud computing