Engineering and Scientific International Journal (ESIJ)
Volume 3, Issue 4, October– December 2016
ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Multilevel Data Security and Revocability in Cloud Storage System

Anoop.M[#1], Arul Jodhi.T[*2]

[1&2]*Assistant Professor, Department of Computer Science,*
Alpha Arts and Science College, Porur, Chennai.
[1]*profanoopcs@rediffmail.com*
[2]*aruljothiapril19.2012@gmail.com*

Abstract— In this paper, we propose a data security protection mechanism with revocability for cloud storage system. An encrypted message is send from the sender to receiver through a cloud storage server. The sender only knows the identity of the receiver. The receiver's certificate or its public key is not known to the sender and also the receiver has two things in order to decrypt the ciphertext. The secret key is stored in the computer and the unique personal security device which connects to the computer. It is impossible to decrypt the cipher text without both the key. Once the Security device is lost, the device is revoked. Since revoking has been done the ciphertext cannot be decrypted. This can be achieved by some algorithms to change the existing ciphertext to be un-decrypt able by this device.

Keywords— *Multilevel, Security, Recoverability, Cloud Storage*

1. Introduction

Cloud storage is a prototype of networked storage system where data is stored in group of storage which are uploaded by third parties [1],[2],[3]. There are many advantages to use cloud storage. The most notable is data accessibility. Data stored in cloud can be accessed at any time from any place as long as there is network access. Another advantage of cloud storage is data sharing between users. If USER A wants to share a piece of data to USER B, it may be difficult to send it by e-mails due to the size of data. Instead, A uploads the file to a cloud storage system so that B can download it at any time.

Among the advantages, outsourcing data storage also increases the attack surface area at the same time. When data is distributed, the unauthorized physical access to the data will be more. By sharing storage and networks with many other user it is also possible for other unauthorized user to access your data. This may be because of bad equipment, mistaken action or sometimes criminal intent.

Encryption technology is one of the promising solutions to offset the risk management in cloud storage. It can protect data as it is transmitted to and from the cloud service. Asymmetric encryption allows the encryptor to use only the public information to generate a ciphertext while the receiver uses their own secret key to decrypt. In a normal asymmetric encryption there is a single key corresponding to a public key. The decryption of ciphertext only requires this key. The key is usually stored inside either a personal computer or a trusted server, and may be protected by a password. While the users are connected with the outside world through internet, the system may suffer from the risk that a third party may intrude into it to compromise the secret key without letting the owner know. The computer storing a user decryption key may be used by another user when the original user is away. Therefore, there exists a need to enhance the security protection.

As cloud computing becomes more mature and there will be more applications and storage services provided by the cloud, it is easy to forecast that the security for data guard in the cloud should be further enhanced [4],[5],[6]. The concept of two factor encryption, which is one of the encryption trends for data protection, has been spread into some real world application (e-banking). However the applications suffer from a potential risk about factor recoverability that may limit their practicability. A flexible and scalable two factor encryption mechanism is really desirable in the era of cloud computing.

2. Literature Survey

2.1 Double Encryption

A need of a secret key and an security device is needed for double encryption purpose. The encryption process is executed twice. First encrypt the plain text using the public key. Second encryption is made by the public key or serial number of the security devices. For decryption the security device first decrypts once and then by using the private key second decryption is done and the plain text is achieved.

The issues in this model are that, if the user has lost his/her security device, then the corresponding ciphertext in the cloud cannot be decrypted forever. There is no recoverability support in this model. Since this is the identity based method and to know the two secret key it is the need to know both the keys and the identity is loses where the sender needs to know not only the identity but also the another serial number.

2.2 Split the Secret Key into Two Parts

Another way used, in which the secret key is split into two parts. The first part is stored in the computer while the

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

second part is embedded into a security device. The security of a normal encryption scheme cannot be guaranteed if part of the secret key has been exposed. We include another security primitive called leakage resilient encryption [9],[8],[7]. This scheme guaranteed the security even if the leakage of the secret key is up to some bits and not the whole secret key. The user needs to obtain a replacement device so that he can continue to decrypt his corresponding secret key. The trivial way is to copy the same bits as in the stolen device to the new device by the private key generator (PKG). The most secure way is to cease the validity of the stolen security device.

The issue in this method is that if the security device is reported as lost, the user can no longer use the old device to login. This using leakage resilient primitive cannot provide this security feature which is considered as the most important criterion of two factor security protection.

2.3 Other Methods

In a druva system, a message is first encrypted under a user key K1, and next uploaded to a cloud server. The user key k1 is encrypted by another user key K2, and stored in the server as well. The key K2 is detained by the user. When regain the message, the user needs to use K2 to recover K1 which is used to retrieve message. This mechanism suffers from a risk in practice: once the user loses the key K2, all data of the user stored in the cloud storage cannot be retrieved. The lack of revocability for encryption factor limits the flexibility of the system.

3. Our Contribution

In this paper, we propose a novel effective data security and revocability mechanism for data stored in the cloud.

- Our system is an Identity Based Encryption (IBE) based mechanism. The sender only needs to know the identity of the receiver in order to send an encrypted data to them. No other information of the receiver is required. Then the sender sends the ciphertext to the cloud where the receiver can download it at any time.
- Our system provides two factor data encryption protection. In order to decrypt the data stored in the cloud, the user needs to possess two things. First, the user needs to have his/her secret key which is stored in the computer. Second, the user needs to have a unique personal security device which will be used to connect to the computer. It is impossible to decrypt the ciphertext without either piece.
- Our system also provides security device revocability. Once the security device is stolen or reported ass lost, this device is revoked. That is using this device can no longer decrypt any ciphertext in any circumstances. The cloud will immediately execute some algorithms to change the existing ciphertext to be un-decryptable this device. While the user needs to use his new/ replacement device to decrypt his/ her ciphertext. This process is completely transparent to the sender.
- The cloud server cannot decrypt any ciphertext at anytime.

3.1 Cryptosystems with Two Secret Keys

There are two kinds of cryptosystems that requires two secret keys for decryption. They are certificate less cryptosystem (CLC) and certificate based cryptosystem. Certificate less cryptosystem [10] combines the merits of identity-based cryptosystem (IBC) and the traditional public-key infrastructure (PKI). In a CLC, a user with an identity chooses their private (own) user secret key and user public key. At the same time the authority called the Key Generation Centre (KGC) further generates a partial secret key according to his identity. Encryption or signature verification requires the knowledge of both the public key and the user identity.

On the opposite, decryption or signature generation requires the knowledge of both the user secret key and the partial secret key given by the KGC. Different from the traditional PKI, there is no certificate required. Thus the costly certificate validation process can be eliminated. However, the encryptor or the signature verifier still needs to know the user public key. It is less convenient than IBC where only identity is required for encryption or signature verification.

Similar to CLC, another primitive called certificate-based cryptosystem (CBC) was introduced. The concept is almost the same as CLC, except that the partial secret key given by the KGC is a signature of the identity and the public key of the user by the KGC.

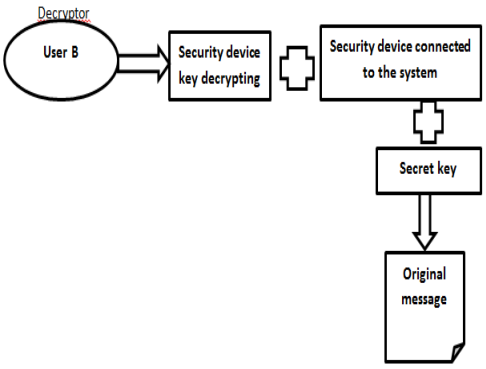

Fig.1: Decryption of a data from the cloud storage

3.2 Cryptosystems with Online Authority

The purpose of mediated cryptography [11] is revocation of public keys. It requires an online mediator, referred to a Security Mediator (SEM), for every transaction. The SEM also provides a control of security capabilities. If the SEM

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

does not cooperate then no transactions with the public key are possible any longer. In other words, any revoked user cannot get the cooperation from the SEM. That means revoked users cannot decrypt any ciphertext successfully.

Later on, security mediated certificateless (SMC) cryptography is introduced in which a user has a secret key, public key and an identity. The user secret key and the SEM are required to decrypt a ciphertext or sign a message. On the opposite side, the user public key and the corresponding identity are needed for signature verification or encryption. Since the SEM is controlled by the revocation authority, the authority can refuse to provide any cooperation for revoked user so that no revoked user can generate signature or decrypt ciphertext.

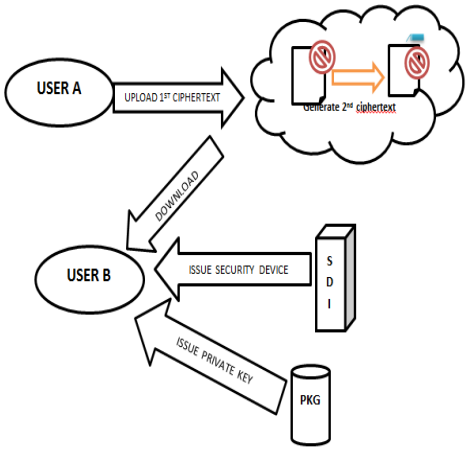

Fig.2: Data uploading and downloading in cloud storage system using double security protection

The main purpose of SMC is to solve the revocation problem. Thus the SME is controlled by the authority and it has to be online for every signature signing and ciphertext decryption. Furthermore, it is not identity-based. The encryptor needs to know the corresponding public key in addition to the identity. That makes the system less practical and loses the advantages of using identity-based system.

3.3 Cryptosystem with Security Device

The paradigm of key-insulated cryptography where [12], there is a physically-secure but computationally-limited device in the system. The lasting key is stored in this tool, while aad-hoc secret key is kept by users on a powerful but diffident device where cryptographic computations occur. Ad-hoc secret keys are then refreshed at various time intervals via interaction between the user and the base while the public key remains constant throughout the lifetime of the system. The user obtains a partial secret key from the device at the beginning of each time period. He then combines this partial secret key with the one from the previous period, in order to renew the secret key for the current time period.

Key-insulated cryptosystem requires all users to update their key in every time period. It may require some costly time synchronization algorithms between users which may not be practical in many scenarios. The key update process requires the security device. Once the key has been updated, the signing or decryption algorithm does not require the device anymore within the same time period. While our concept does require the security device every time the user tries to decrypt the ciphertext. Furthermore, there is no key updating required in our system. Thus we do not require any synchronization within the whole system.

3.4 Cryptosystem with Revocability

We introduce IBE-based systems supporting revocability. The first revocable IBE is proposed by Boneh and Franklin [8], in which a ciphertext is encrypted under an identity id and a time period T, and a non-revoked user is issued a private key skid; T by a PKG such that the user can access the data in T.Boldyreva, Goyal and Kumar proposed the security notion for revocable IBE. To achieve adaptive security, Libert and Vergnaud [15] proposed a revocable IBE scheme based on the combination of attribute-based encryption and IBE. Recently, Seo and Emura formalized a revised notion for revocable IBE. Since its introduction, there are many variants of revocable IBE, such as [17]. The premise of a revocable IBE system is mainly related to a time period: next the decryption rights of the next time period relies on a secret token (for the next time period) issued by PKG and a current time period key. However, this premise yields inconvenience once the current time period key is lost.

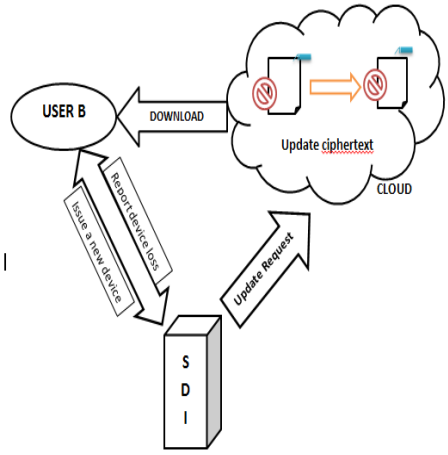

Fig.3: Revocability of the data from the cloud storage system

Another cryptosystem supporting revocability is proxy re-encryption (PRE). Decryption rights delegation is introduced in [16]. Blaze, Bleumer and Strauss [13] formally defined the notion of PRE. To employ PRE in the IBE setting, Green and Ateniese [14] defined the notion of identity based PRE (IB-PRE). Later on, Tang,Hartel and

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Jonker proposed a CPA-secure IB-PRE scheme, in which delegator and delegate can belong to different domains. Among of the previously introduced IB-PRE systems, [14] is the most efficient one without loss of revocability. We state that leveraging [14] can only achieve one of our design goals, revocability, but not two-factor protection

4. Our Intuition

In our system, we have the following entities:

A. Private Key Generator: It is a trusted party responsible for issuing private key of every user.

B. Security Device Issuer (SDI): It is a trusted party responsible for issuing security device of every user.

C. User A: He/she is the sender of the ciphertext. He /She only knows the identity (e.g., email address) of the receiver but nothing else related to the receiver. After she has created the ciphertext, she sends to the cloud server to let the receiver for download.

D. User B: He/she is the receiver of the ciphertext and has a unique identity (e.g., email address). The ciphertext is stored on cloud storage while he can download it for decryption. He has a private key (stored in his computer) and a security device (that contains some secret information related to his identity). They are given by the PKG. The decryption of ciphertext requires both the private key and the security device.

E. Cloud Server: The cloud server is responsible for storing all ciphertext (for receiver to download). Once a user has reported lost of his security device (and has obtained a new one from the PKG), the cloud acts as a proxy to re-encrypt all his past and future ciphertext corresponding to the new device. That is, the old device is revoked.

When a new system user B, joins our system, a PKG will issue a private key, and SDI will issue a security device to him. Both the private key and the security device are necessary for recovering a data from its encrypted format. In ordinary data sharing, a data sender, say user A, first encrypts the sharing data under the identity of a data receiver, say user B, and next uploads the ciphertext to the cloud server. Here we refer to this ciphertext as first-level ciphertext. After receiving the first-level ciphertext from user A, the cloud server then turns the ciphertext to become a second-level ciphertext for the corresponding security device belonging to user B. user B then downloads the second level ciphertext from the cloud, and next recovers the data from its encrypted form by using his private key and security device.

When the security device of user B is either lost or stolen, user B first reports the issue to the SDI. The SDI then issues a new security device to user B, and meanwhile, it sends a request of updating user B's corresponding ciphertext along with a special key to the cloud server. The cloud server updates the ciphertexts of user B under an old security device to the ones under a new device. However, it does not gain access to the underlying data in the update process. Here user B is allowed to download and recover the data by using his private key and new security device.

5. Construction

We use two different encryption technologies: one is IBE and the other is traditional Public Key Encryption (PKE). We first allow a user to generate a first level ciphertext under a receiver's identity. The first level ciphertext will be further transformed into a second level ciphertext corresponding to a security device. The resulting ciphertext can be decrypted by a valid receiver with secret key and security device. Here, one might doubt that our construction is a trivial and straightforward combination of two different encryptions. Unfortunately, this is not true due to the fact that we need to further support security device revocability. A trivial combination of IBE and PKE cannot achieve our goal. To support revocability, we employ re-encryption technology such that the part of ciphertext for an old security device can be updated for a new device if the old device is revoked. Meanwhile, we need to generate a special key for the above ciphertext conversion. We also guarantee that the cloud server cannot achieve any knowledge of message by accessing the special key, the old ciphertext and the updated ciphertext.

A. Setup phase: The setup phase generates all public parameters and master secret key used throughout the execution of system. The public parameters are shared with all parties participating into the system (including data sender/receiver, cloud server and a PKG), while the master secret key is given to the PKG.

B. Key and device issued phase: A SDI and a PKG will respectively generate a security device and a secret key for a registered user in secure channel such that the user can combine the security device with the secret key to recover message from its encrypted format. The SDI chooses and sets the security device's description information and its corresponding secret information. The SDI finally delivers the security device to a user ID. The SDI stores the tuple in a list shared with the cloud storage system. The PKG sets the secret key for a user ID.

C. First-level ciphertext generation phase: A data sender encrypts a data under the identity of a data receiver, and further sends the encrypted data to the cloud server. Knowing public parameters param, a data and a receiver's identity, a data sender encrypts a data to a first level encryption.

D. Second-level ciphertext phase: After receiving the first level ciphertext of a data from the data sender, the cloud server generates the second-level ciphertext. Knowing public parameters param, a first level encryption for the user, and the information stored in List, the cloud server encrypts second-level ciphertext.

E. Device updated phase: Once a device of a user needs to be updated due to some incidences (e.g., it is either lost or stolen), the user first reports the issue to the SDI. The SDI

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

then issues a new device for the user. The SDI chooses and sets the security device's description information and its corresponding secret information. The SDI finally delivers the security device to a user ID and updates the list.

F. Ciphertext updated phase: The SDI notifies the cloud server to update the ciphertext of the user by sending a special piece of information. The SDI first sends a piece of information to the cloud server so as to inform the cloud to execute the ciphertext updated process. After receiving the information, the cloud server updates the ciphertext.

G. Data recovery phase: A data receiver uses a decryption key and a device to recover the data.

6. Conclusions

In this paper, we introduced a novel two-agent data bulwark mechanism for cloud storage system, in which a data shipper is allowed to encrypt the data with knowledge of the identity of a beneficiary only, while the beneficiary is required to use both his/her secret key and a security device to gain entree to the data. Our solution not only enhances the confidentiality of the data, but also offers the revocability of the device so that once the mechanism is revoked, the equivalent ciphertext will be updated automatically by the cloud server without any notice of the data owner. Furthermore, we demonstrated the security evidence and efficiency analysis for our system.

7. Future Enhancement

In this paper, we consider the following threats: Decrypt without security device: The adversary tries to decrypt the ciphertext without the security device, or using a revoked security device, or using another security device belonging to others. It can have its own secret key. Decrypt without secret key: The adversary tries to decrypt the ciphertext without any secret key. It can have its own security device .Note that the above threat model has already captured the semi-trust behaviour of the cloud server.

References

[1] H. C. H. Chen, Y. Hu, P. P. C. Lee, and Y. Tang, "NCCloud: A network-coding-based storage system in a cloud-of-clouds," IEEE Trans. Comput., vol. 63, no. 1, Jan (2014) pp. 31–44,.

[2] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," IEEE Trans.Comput., vol. 62, no. 2, Feb.(2013)pp. 362–375.

[3] H. Wang, "Proxy provable data possession in public clouds," IEEE Trans. Services Comput., vol. 6, no. 4, pp. 551–559, Oct.–Dec. 2013.

[4] C.-K. Chu, S. S. M. Chow, W.-G.Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, Feb. (2014)pp. 468– 477.

[5] L. Ferretti, M. Colajanni, and M. Marchetti, "Distributed, concurrent, and independent access to encrypted cloud databases," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, Feb(2014) pp. 437–446.

[6] V. Varadharajan and U. K. Tupakula, "Security as a service model for cloud environment," IEEE Trans. Netw. Service Manage., vol. 11, no. 1, Mar (2014) pp. 60–75.

[7] M. Naor and G. Segev, "Public-key cryptosystems resilient to key leakage," in Proc. 29th Annu. Int. Cryptol. Conf.,(2009) pp. 18–35.

[8] Y. Dodis, Y. T. Kalai, and S. Lovett, "On cryptography with auxiliary input," Proc. 41 Annu.ACMSymp. Theory Comput., (2009), pp. 621–630.

[9] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in Proc. 6th Theory Cryptography Conf., (2009), pp. 474–495.

[10] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key crypto.,"Proc.9th Conf.Theory Appl.Cryptol.,(2003), pp. 452–73.

[11] D. Boneh, X. Ding, and G. Tsudik, "Fine-grained control of security capabilities," ACM Trans.Inte. Techn., vol.4, no.1,(2004) pp. 60– 82.

[12] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Strong key-insulated signature schemes," in Proc. Int. Conf. Theory Appl. Cryptographic Techn., (2003) pp. 130–144.

[13] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in Proc. Int. Conf. Theory Appl. Cryptographic Techn., (1998) pp. 127–144.

[14] M. Green and G. Ateniese, "Identity-based proxy re-encryption," Proc. 5th Int. Conf. Appl. Crypt. Netw. Security, (2007) pp.288–306.

[15] B. Libert and D. Vergnaud, Adaptive-id secure revocable identitybased encryption, in Proc. Cryptographers Track RSA Conf., (2009) pp. 1–15.

[16] M. Mambo and E. Okamoto, "Proxy cryptosystems: Delegation of the power to decrypt ciphertexts," IEICE Trans., vol. E80-A, no. 1, (1997)pp. 54–63.

[17] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in Proc. 32nd Annu.Cryptol. Conf. Adv. Cryptol., (2012,)pp. 199–217.