ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

EFFICIENT CERTIFICATE BASED ENCRYPTION IN THE STANDARD MODEL IN CLOUD STORAGE

Gangalam Sarika
Sr. Systems Engineer, Cognizant Technology Solutions, Hyderabad.

Abstract-: In this paper, we propose a two-factor data security protection mechanism with factor revocability for cloud storage system. Our system allows a sender to send an encrypted message to a receiver through a cloud storage server. The sender only needs to know the identity of the receiver but no other information (such as its public key or its certificate). The receiver needs to possess two things in order to decrypt the cipher text. The first thing is his/her secret key stored in the computer. The second thing is a unique personal security device which connects to the computer. It is impossible to decrypt the cipher text without either piece. More importantly, once the security device is stolen or lost, this device is revoked. It cannot be used to decrypt any cipher text. This can be done by the cloud server which will immediately execute some algorithms to change the existing cipher text to be un-decryptable by this device. This process is completely transparent to the sender. Furthermore, the cloud server cannot decrypt any cipher text at any time. The security and efficiency analysis show that our system is not only secure but also practical.

Keywords: Cloud Computing, Cloud Storage, Encryption Model.

I.Introduction

Cloud storage is a model of networked storage system where data is stored in pools of storage which are generally hosted by third parties. There are many benefits to use cloud storage. The most notable is data accessibility. Data stored in the cloud can be accessed at any time from any place as long as there is network access. Storage maintenance tasks, such as purchasing additional storage capacity, can be offloaded to the responsibility of a service provider. Another advantage of cloud storage is data sharing between users. If Alice wants to share a piece of data (e.g. a video) to Bob, it may be difficult for her to send it by email due to the size of data. Instead, Alice uploads the file to a cloud storage system so that Bob can download it at anytime. Despite its advantages, outsourcing data storage also increases the attack surface area at the same time. For example, when data is distributed, the more locations it is stored the higher risk it contains for unauthorized physical access to the data. By sharing storage and networks with many other users it is also possible for other unauthorized users to access your data. This may be due to mistaken actions, faulty equipment, or sometimes because of criminal intent. A promising solution to offset the risk is to deploy encryption technology. Encryption can protect data as it is being transmitted to and from the cloud service. It can further protect data that is stored at the service provider. Even there is an unauthorized adversary who has gained usage of computers is also common. For example, in a college, a public computer in a copier room will be shared with all students staying at the same floor. In these cases, the secret key can be compromised by some attackers who can access the victim's personal data stored in the cloud access to the cloud, as the data has been encrypted, the adversary cannot get any information about the plaintext. Asymmetric encryption allows the encryptor to use only the public information (e.g. public key or identity of the receiver) to generate a cipher text while the receiver uses his/her own secret key to decrypt. This is the most convenient mode of encryption for data transition, due to the elimination of key management existed in symmetric encryption.

Enhanced Security Protection

In a normal asymmetric encryption, there is a single secret key corresponding to a public key or an identity. The decryption of cipher text 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. The security protection is sufficient if the computer/server is isolated from an opening network. Unfortunately, this is not what happens in the real life. When being connected with the world through the Internet, the computer/server may suffer from a potential risk that hackers may intrude into it to compromise the secret key without letting the key owner know. In the physical security aspect, the computer storing a user decryption key may be used by another user when the original computer user (i.e. the key owner) is away (e.g. when the user goes to toilet for a while without locking the machine). In an enterprise or college, the sharing

system. Therefore, there exists a need to enhance IEEE Transactions on Computers (Volume: 65, Issue: 6, June 1 2016),30 July 2016 2 the security protection. An analogy is e-banking security. Many e-banking applications require a user to use both a password and a security device

[1]Corresponding Author

(two factors) to login system for money transfer. The security device may display a one-time password to let the user type it into the system, or it may be needed to connect with the computer (e.g. through USB or NFC). The purpose of using two factors is to enhance the security protection for the access control. As cloud computing becomes more mature and there will be more applications and storage services provided by the cloud, it is easy to foresee that the security for data protection in the cloud should be further enhanced. They will become more sensitive and important, as if the e-banking analogy. Actually, we have noticed that the concept of two-factor encryption, which is one of the encryption trends for data protection1 , has been spread into some real-world applications, for example, full disk encryption with Ubuntu system, AT&T two factor encryption for Smartphones2 , electronic vaulting and druva - cloud-based data encryption3 . However, these applications suffer from a potential risk about factor revocability that may limit their practicability. Note we will explain it later. A flexible and scalable two factor encryption mechanism is really desirable in the era of cloud computing. That motivates our work.

2.1. Some Naive Approaches We discuss some naive approaches for enhancement of security protection and explain why they are not the best candidate to achieve the goal of flexibility. 1) Double encryption: A security device (with an additional public key or serial number) is still required. The encryption process is executed twice. First encrypt the plaintext corresponding to the public key or identity of the user. Then encrypt it again corresponding to the public key or serial number of the security device. For the decryption stage, the security device first decrypts once. The partially decrypted cipher text is then passed to the computer which uses the user secret key to further decrypt it. Without either part (user secret key or security device) one cannot decrypt the cipher text. It seems that this naive approach can achieve our goal. However, there exist many practical issues that it cannot solve. For example, • If the user has lost his security device, then his/her corresponding cipher text in the cloud cannot be decrypted forever! That is, the approach cannot support security device update/revocability.

• The sender needs to know the serial number / public key of the security device, in additional to the user's identity / public key. That makes the encryption process more complicated. In the case of identity-based encryption, the concept of "identity-based" has been totally lost as the sender needs to know not only the identity but another serial number!

Split the secret key into two parts: Another naive way to think of is to simply split the secret key into two parts. The first part is stored in the computer while the second part is embedded into a security device. Similar to the above approach, without either part one cannot decrypt the cipher text. Again it seems that this approach can achieve our goal. However, note that the security of a normal encryption scheme cannot be guaranteed if part of the secret key has been exposed. The security is only guaranteed if the whole secret key has not been exposed to the adversary. In other words, if we simply split the secret key into two parts, the adversary with either part may have non-negligible chance to decrypt (or at least to know some information about the plaintext). This is not the case that we expect. There exists another cryptographic primitive called "leakage-resilient encryption" [1]. The security of the scheme is still guaranteed if the leakage of the secret key is up to certain bits such that the knowledge of these bits does not help to recover the whole secret key. However, though using leakage resilient primitive can safeguard the leakage of certain bits, there exists another practical limitation. Suppose we put part of the secret key into the security device. Unfortunately the device is stolen. 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). This approach can be easily achieved. Nevertheless, there exists security risk. If the adversary (who has stolen the security device) can also break into the computer where the other part of secret key is stored, then it can decrypt all cipher text corresponding to the victim user. The most secure way is to cease the validity of the stolen security device. The same analogy is the online banking. A user needs to have a security device (together with the knowledge of his/her password) in order to login the e-banking service. If the security device is reported as lost, the user can no longer use the old device to login. Thus using leakage resilient primitive cannot provide this security feature which is considered as the most important criterion of two factor security protection. 3) Other methods: Some real-world systems, such as AT&T and druva, also leverage two-factor en- IEEE Transactions on Computers (Volume: 65, Issue: 6, June 1 2016),30 July 2016 3 encryption techniques to protect message from being leaked to malicious users. However, their techniques suffer from a potential practical risk. Below we take druva system as an example. 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 further encrypted by another user key k2, and stored in the server as well. The key k2 is held by the user. When retrieving the message, the user needs to use k2 to recover k1 which is further used to recover m. It is undeniable that this message-key encrypt mechanism is much better than the mode only using a single key to encrypt an outsourced data, and storing the cipher text along with the key in the server. Nevertheless, this mechanism suffers from a potential risk in practice (which we have mentioned previously): once the user loses the key k2, all data of the user stored in the cloud cannot be retrieved.

The lack of revocability for encryption factor limits the flexibility of the system. 1.2 Our Contributions In this paper, we propose a novel two-factor security protection mechanism for data stored in the cloud. Our mechanism provides the following nice features: 1) Our system is an IBE (Identity-based encryption)- based mechanism. That is, the sender only needs to know the identity of the receiver in order to send an encrypted data (cipher text) to him/her. No other information of the receiver (e.g. public key, certificate etc.) is required. Then the sender sends the cipher text to the cloud where the receiver can download it at anytime. 2) 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 (e.g. USB, Bluetooth and NFC). It is impossible to decrypt the cipher text without either piece. 3) More importantly, our system, for the first time, provides security device (one of the factors) revocability. Once the security device is stolen or reported as lost, this device is revoked. That is, using this device can no longer decrypt any cipher text (corresponding to the user) in any circumstance. The cloud will immediately execute some algorithms to change the existing cipher text to be un-decryptable by this device. While the user needs to use his new / replacement device (together with his secret key) to decrypt his/her cipher text. This process is completely transparent to the sender. 4) The cloud server cannot decrypt any cipher text at any time. We provide an estimation of the running time of our prototype to show its practicality, using some benchmark results. We also note that although there exist some naive approaches that seem to achieve our goal, we have discussed in Section 1.1 that there are many limitations by each of them and thus we believe our mechanism is the first to achieve all the above mentioned features in the literature.

II.Related Work

We first review some solutions which may contain similar functionalities. We will further explain why they cannot fully achieve our goal. 2.1 Cryptosystems with Two Secret Keys There are two kinds of cryptosystems that requires two secret keys for decryption. They are certificate lesscryptosystem and certificate-based cryptosystem. Certificate less cryptosystem (CLC) was first introduced in [2] and further improvements can be found in [4]. It combines the merits of identity based cryptosystem (IBC) and the traditional public-key infrastructure (PKI). In a CLC, a user with an identity chooses his 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. Further variants may include [3]. The concept is almost the same as CLC, except that the partial secret key given by the KGC (which is called the certificate) is a signature of the identity and the public key of the user by the KGC. (Note that in CLC, the partial secret key given by the KGC is just the signature of the identity of the user.) Due to the similarities, CBC faces the same disadvantages as CLC mentioned above.

2.2. Cryptosystems with Online Authority Mediated cryptography was first introduced in for the purpose of revocation of public keys. It requires an online mediator, referred to a SEM (SEcurity Mediator), for every transaction. The SEM also provides a control of security capabilities. If the SEM does not cooperate then no transactions with the public key are possible any longer. In other words, any revoked user cannot get the IEEE Transactions on Computers (Volume: 65, Issue: 6, June 1 2016),30 July 2016 4 cooperation from the SEM. That means revoked users cannot decrypt any cipher text successfully. Later on, this notion was further generalized as security mediated certificate less (SMC) cryptography. In a SMC system, a user has a secret key, public key and an identity. The user secret key and the SEM are required to decrypt a cipher text 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 cipher text. Note that SMC is different from our concept. 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 cipher text decryption. Furthermore, it is not identity-based. The encryptor (or signature verifier) needs to know the corresponding public key in addition to the identity. That makes the system less practical and looses the advantages of using identity-based system.

2.3 Cryptosystem with Security Device The paradigm of key-insulated cryptography was introduced in and variants were proposed in [17], [22], [25], [32]. There is a physically-secure but computationally-limited device in the system. A long-term key is stored in this device, while a short-term secret key is kept by users on a powerful but insecure device where cryptographic computations take place. Short term secrets are then refreshed at discrete time periods via interaction between the user and the base while the public key remains unchanged 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. Different from our concept, 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 cipher text. Furthermore, there is no key updating required in our system. Thus we do not require any synchronization within the whole system. 2.4 Cryptosystem with Revocability Since our system is an IBE-based mechanism, we below introduce IBE-based systems supporting revocability. The first revocable IBE is proposed by Boneh and Franklin [8], in which a cipher text is encrypted under an identity id and a time period T, and a nonrevoked 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 [6] proposed the security notion for revocable IBE. To achieve adaptive security, Libert and Vergnaud [26] proposed a revocable IBE scheme based on the combination of attribute-based encryption and IBE. Recently, Seo and Emura [39] formalized a revised notion for revocable IBE. Since its introduction, there are many variants of revocable IBE, such as [38].

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. Another cryptosystem supporting revocability is proxy re-encryption (PRE). Decryption rights delegation is introduced in [35]. Blaze, Bleumer and Strauss [5] formally defined the notion of PRE. To employ PRE in the IBE setting, Green and Ateniese [20] defined the notion of identity-based PRE (IB-PRE). Later on, Tang, Hartel and Jonker [41] proposed a CPA-secure IB-PRE scheme, in which delegator and delegatee can belong to different domains. After that there are many IB-PRE systems have been proposed to support different user requirements. Among of the previously introduced IB-PRE systems, [20] is the most efficient one without loss of revocability. We state that leveraging [20] can only achieve one of our design goals, revocability, but not two-factor protection.

III.Conclusion

In this paper, we introduced a novel two-factor data security protection mechanism for cloud storage system, in which a data sender is allowed to encrypt the data with knowledge of the identity of a receiver only, while the receiver is required to use both his/her secret key and a security device to gain access 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 device is revoked, the corresponding cipher text will be updated automatically by the cloud server without any notice of the data owner. Furthermore, we presented the security proof and efficiency analysis for our system.

References

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan.Simultaneous hardcore bits and cryptography against memory attacks. In TCC, volume 5444 of Lecture Notes in Computer Science, pages 474–495. Springer, 2009.

[2] S. S. Al-Riyami and K. G. Paterson.Certificate less public key cryptography. In ASIACRYPT, volume 2894 of Lecture Notes in Computer Science, pages 452–473. Springer, 2003. [3] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen. Certificate based (linkable) ring signature. In ISPEC, volume 4464 of Lecture Notes in Computer Science, pages 79–92. Springer, 2007.

[4] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang. Malicious kgc attacks in certificate less cryptography. In ASIACCS, pages 302–311. ACM, 2007.

[5] M. Blaze, G. Bleumer, and M. Strauss.Divertible protocols and atomic proxy cryptography. In K. Nyberg, editor, EUROCRYPT, volume 1403 of LNCS, pages 127–144. Springer, 1998.

[6] A. Boldyreva, V. Goyal, and V. Kumar.Identity-based encryption with efficient revocation. In P. Ning, P. F. Syverson, and S. Jha, editors, ACM Conference on Computer and Communications Security, pages 417–426. ACM, 2008.