International Journal for Modern Trends in Science and Technology
Volume: 03, Issue No: 01, January 2017
ISSN: 2455-3778
http://www.ijmtst.com

Time Domain Attribute Base Access Control for Cloud Based Content Sharing: A Cryptographic Approach

Prof. Rucha Dixit [1] | Shubham Shivathare [2] | Gaurav Ganesh [3]

[1]Department of Computer Engineering, JSCOE, Pune, Maharashtra, India
[2]Department of Computer Engineering, JSCOE, Pune, Maharashtra, India
[3]Department of Computer Engineering, JSCOE, Pune, Maharashtra, India

To Cite this Article
Prof. Rucha Dixit, Shubham Shivathare and Gaurav Ganesh, "Time Domain Attribute Base Access Control For Cloud Based Content Sharing: A Cryptoghaphic Approach", International Journal for Modern Trends in Science and Technology, Vol. 03, Issue 01, 2017, pp. 74-78.

ABSTRACT

In the past few years, the rapid development of cloud storage services makes it easier than ever for cloud users to share data among themselves. To ensure users' confidence of the integrity of their shared data on cloud, a number of techniques have been introduced for data integrity auditing which focuses on various practical features, secure data destructing, public integrity auditing etc. To overcome the problem self-destruction method is proposed. All the data and their copies become self-destructed after user specified time period. After user specified time period key should be destructed or become unreadable. The file should be encrypted before upload and decrypted before download. Any user can download file till the timeout. Self-destruction mechanism reduces the time taken to upload and download file as compared to native system. In this we are using a key-policy attribute based encryption with time-specified attributes (KPTSABE), a novel secure data self-destructing scheme in cloud computing. In the KP-TSABE scheme, every cipher text is labeled with a time interval while private key is related with a time instant. The sensitive data will be securely self-destructed after a user-specified expiration time. Comprehensive comparisons of the security properties indicate that the KP-TSABE scheme fulfills the security requirements and is superior to other existing schemes.

KEYWORDS: *Cloud Computing, Cryptography, Self-Destruction, Data Privacy, Sensitive Data*

I. INTRODUCTION

Cloud computing is considered as the next step in the evolution of on-demand information technology which combines a set of existing and new techniques from research areas such as service-oriented architectures (SOA) and virtualization. With the rapid development of different cloud computing technology and services, it is routine for user to leverage cloud storage services to share data with others in a friend circle, e.g. Dropbox, Google Drive and Ali Cloud The shared data in cloud servers, however, usually contains users' sensitive information (e.g., personal profile, financial information, health records, etc.) and needs to be well protected.

As the ownership of the data is separated from the administration of them the cloud servers may move user data to other cloud servers in outsourcing or share them in cloud searching. Therefore, it becomes a huge challenge to secure the privacy of those shared data in cloud, especially in cross-cloud and big data environment. In order to meet this challenge, it is necessary to design a

comprehensive solution to support user-defined authorization time and to provide fine-grained access control during this period. The shared data should be self-destroyed after the user-defined expiration period.

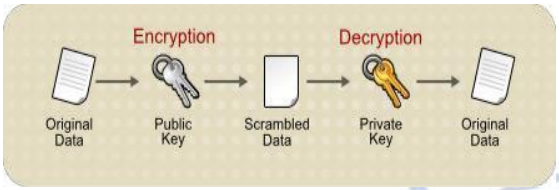

Figure 1

One of the methods to alleviate the problems is to collect data as a common encrypted form. The drawback of encrypting data is that the user cannot share his/her encrypted data at a fine-grained level. When a data owner wants to share someone his/her information, the owner must know exactly the one he/she wants to share with. In many applications, the data owner wants to share information with most of users according to the security policy based on the users' credentials.

Attribute based encryption (ABE) has excellent advantages based on the old public key encryption instead of one-to-one encryption because it achieves flexible one-to-many encryption. ABE scheme provides a powerful method to achieve both data security and fine-grained access control. In the key-policy ABE (KP-ABE) scheme to be elaborated in this paper, the ciphertext is labeled with number of set of descriptive attributes. Only when the set of descriptive attributes satisfies the access structure in the key, the user can get the plaintext.

In KP-ABE, the idea is reversed: the ciphertext contains a set of attributes and the private key is related to the access structure. The first construction of KP-ABE scheme was proposed. In their scheme, when a user made a private request, the faithful authority determined which combination of attributes must appear in the ciphertext for the user to decrypt. Instead of using the Shamir secret key technique in the private key, this scheme used a more generalized form of secret sharing to enforce a monotonic access tree

II. LITERATURE SURVEY

In several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Attribute-based encryption is a type of public-key encryption in which the secret key of a user and the ciphertext are dependent upon attributes (e.g. the country in which he lives, or the kind of subscription he has). In such a system, the decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext. A crucial security aspect of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access. Attribute-based encryption (ABE) is a vision of public key encryption that allows users to encrypt and decrypt messages based on user attributes. This functionality comes at a cost. In a typical implementation, the size of the ciphertext is proportional to the number of attributes associated with it and the decryption time is proportional to the number of attributes used during decryption. Specifically, many practical ABE implementations require one pairing operation per attribute used during decryption.

In this work we present a new Attribute-Based Encryption scheme where private keys can represent any access formula over attributes, including non-monotone ones. The widespread availability of networks, such as the Internet, has prompted a proliferation of both stationary and mobile devices capable of sharing and accessing data across networks spanning multiple administrative domains. Today, efficient data storage is vital for almost every scientific, academic, or business organization. Cryptographic file systems address the first problem. These file systems essentially maintain the confidentiality and integrity of the file data by storing it in an encrypted format at the SSPs. With the advent of high speed hardware for encrypting and decrypting data, the overhead in a cryptographic file system due to file encryption and decryption is affordably small. Access control in a cryptographic file system translates into a secure key management problem. Cryptographic access control is achieved by distributing a file's encryption key to only those users that are authorized to access that file.

Public-key encryption (also called asymmetric encryption) involves a pair of keys, a public key and a private key, associated with an entity. The scheme allows public keys to be freely distributed, while only authorized people are able to read data encrypted using this key. In general, to send encrypted data, the data is encrypted with that person's public key, and the person receiving the encrypted data decrypts it with the corresponding

private key. Compared with symmetric-key encryption, public-key encryption requires more processing and may not be feasible for encrypting and decrypting large amounts of data. However, it is possible to use public-key encryption to send a symmetric key, which can then be used to encrypt additional data. This is the approach used by the SSL/TLS protocols. The reverse of the scheme shown in also works: data encrypted with a private key can be decrypted only with the corresponding public key. This is not a recommended practice to encrypt sensitive data, however, because it means that anyone with the public key, which is by definition published, could decrypt the data. Nevertheless, private-key encryption is useful because it means the private key can be used to sign data with a digital signature, an important requirement for electronic commerce and other commercial applications of cryptography. Client software such as Mozilla Firefox can then use the public key to confirm that the message was signed with the appropriate private key and that it has not been tampered with since being signed.

In this paper we propose a full lifecycle privacy protection scheme for sensitive data (FullPP), which is based on identity-based timed-release encryption (ID-TRE) algorithm which encrypts and decrypts key. After generation of key it combines key with encrypted cipher text and pass DHT. DHT shares these cipher text over the network in cloud computing.

This paper addresses this challenging open issue by, on one hand, defining and enforcing access policies based on data attributes, and, on the other hand, allowing the data owner to delegate most of the computation tasks involved in fine grained data access control to entrusted cloud servers without disclosing the underlying data contents. Existing work can be found in the areas of shared cryptographic file systems and access control of outsourced data.

Policy-based file assured deletion is developed upon conventional cryptographic techniques, i.e. it encrypts the outsourced data files to guarantee their confidentiality and integrity, assuredly deletes the file to make them unrecoverable upon request of delete with revocation of file access policies. FADE divides the management of encrypted data and management of encryption keys. The encrypted data remains on the entrusted cloud storage, while encryption keys were maintained by a separate key manager which follows a quorum scheme.

When data is being processed, transformed and stored by the current computer system or network, systems or network must cache, copy or archive it. These copies are essential for systems and the network. As people have no knowledge about these copies and cannot control them, so these copies may leak their privacy. On the other hand, their privacy also can be leaked via Cloud Service Providers negligence, hackers' intrusion or some legal actions. These problems present formidable challenges to protect people's privacy. Personal data stored in the Cloud may contain account numbers, passwords, notes, and other important information that could be used and misused by a miscreant, a competitor, or a court of law. These data are cached, copied, and archived by Cloud Service Providers, often without users' authorization and control. Self destructing data mainly aims at protecting the user data's privacy. All the data and their copies become destructed or unreadable after a user- specified time, without any user intervention. In addition, the decryption key is destructed after the user-specified time. Self destruction is implemented by encrypting data with a key and then retrieving the information needed to reconstruct the decryption key with one or more third parties. Assuming that the key reconstruction information disappears from the retrieval with trust from third parties at the intended time, encrypted data will become permanently unreadable. Even if an attacker retrieve a copy of the encrypted data and the user's cryptographic keys and passphrases after the timeout. Without the user or user's agent taking any precise action to delete it. With no need to alter any stored or archived copies of that data.

III. ADVANTAGES

1) Encryption helps move to the cloud.
2) When you own the keys, You can easily Decommission/Deprovision.
3) Encryption Helps Achieve Secure Multi-Tenancy in the Cloud.
4) Encryption Key Services Prevent Service Providers from Accessing my Data.
5) Encryption Provides Safe Harbor from Breach Notification.
6) Services Providers a Competitive Edge Encryption Gives.
7) As a cloud service provider, you are a guardian of your customers' applications and data. Thieves are getting smarter and regulations are getting more stringent. The good news is that security technology is

also getting better.

8) Encryption and key management software, designed specifically for virtualized environments, can help you significantly improve your security posture, attract new customers, and expand your business with existing clients.
9) Gain competitive advantage and differentiation.
10) Expand revenue potential to customers with sensitive or regulated data
11) Protect customer data against access by unauthorized users.
12) Satisfy data residency and privacy requirements.
13) Reduce hardware costs through cryptographic multi-tenancy.

III. Discussion

Based on the analysis and study of different techniques of in cloud computing, it was found that using self destruction scheme and having KP-TSABE algorithm as a classifier provided the most accurate results compared to other techniques that were studied upon. Another technique wherein statistical analysis was carried out with the MD5 method was also found to be a better technique comparatively.

IV. CONCLUSION

In this paper, we are developing the system for securely data storage facility using cryptographic techniques, we are using hybrid cloud for self destruction feature generating and sharing of key is done for the data sharing purpose so this system is more secure for large data storages.

Acknowledgment

We express true sense of gratitude towards our project guide Prof. Rucha Dixit, Assistant Professor Computer Department for her invaluable co-operation and guidance that she gave us throughout our Project. We specially thank our project coordinator Prof. M. V. Pawar for inspiring us and for providing us all the lab facilities. We would also like to express our appreciation and thanks to HOD Prof. H. A. Hingoliwala and Principal Dr. M. G. Jadhav and all our friends who have assisted us throughout our hard work.

References

[1] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. S. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," Peer to-Peer Networking and Applications.
[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for ne-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and Communications Security. ACM, 2006, pp. 89–98.
[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efcient, and provably secure realization," Public Key Cryptography–PKC 2011, pp. 53–70, 2011.
[4] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and ne-grained data access control in cloud computing," in Proceedings of the 29th IEEE International Conference on Computer Communications. IEEE, 2010, pp. 1–9.
[5] J. H. Cheon, N. Hopper, Y. Kim, and I. Osipkov, "Provably secure timed-release public key encryption," ACM Transactions on Information and System Security (TISSEC), vol. 11, no. 2, p. 4, 2008.
[6] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in Advances in Cryptology–EUROCRYPT 2005. Springer, 2005, pp. 440– 456.
[7] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efcient and privacy-preserving computing in big data era," Network, IEEE, vol. 28, no. 4, pp. 46–50, 2014.
[8] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in Proceedings of the 14th ACM Conference on Computer and Communications Security. ACM, 2007, pp. 195–203.
[9] K. Kasamatsu, T. Matsuda, K. Emura, N. Attrapadung, G. Hanaoka, and H. Imai, "Time-specic encryption from forward-secure encryption," in Security and Cryptography for Networks. Springer, 2012, pp. 184–204.
[10] C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti, "Policybased secure deletion," in Proceedings of the ACM Conference Computer and Communications Security. ACM, 2013, pp. 152–167.
[11] K. G. Paterson and E. A. Quaglia, "Time-specific encryption," in Security and Cryptography for Networks. Springer, 2010, pp. 1–16.
[12] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
[13] A. W. Dent and Q. Tang, "Revisiting the security model for timed-release encryption with pre-open capability," in Proceedings of the Information Security. Springer, 2007, pp. 158–174.
[14] R. Kikuchi, A. Fujioka, Y. Okamoto, and T. Saito, "Strong security notions for timed-release public-key encryption revisited," in Proceedings of the Information Security and Cryptology. Springer, 2012, pp. 88–108.
[15] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, "Secure data deletion from persistent media," in Proceedings of the 2013 AC Conference on Computer and Communications Security. ACM, 2013, pp. 271–284.
[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology–

EUROCRYPT 2005, ser. LNCS, vol. 7371. Springer, 2005, pp. 457–473.

[17] R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in Proceedings of the 18th USENIX Security Symposium, 2009, pp. 299–315.

[18] L. Zeng, S. Chen, Q. Wei, and D. Feng, "Sedas: A selfdestructing data system based on active storage framework," IEEE Transactions on Magnetics, vol. 49, no. 6, pp. 2548–2554, 2013.