MODELLING RE-ENCRYPTION SCHEME FOR SECURE DATA SHARING IN PUBLIC CLOUD

[1]SAIKIRAN CHEBIYYAM, [2]NIKHITHA KAZA, [3]K.VENKATESWARLU
[1,2]U.G Student, Department of Computer Science, GITAM University.
[3]Asst. Professor, Department of Computer Science, GITAM University.
Email: [1]saikiran5717@gmail.com, [2]nikhitha.kaza@gmail.com, [3]venkat.cst@gmail.com**,**

Abstract: A re-encryption scheme of public key cryptography is proposed for secure data sharing in public cloud. The re-encryption scheme offers owner to securely store his data in cloud by encrypting the data over a symmetric DEK and symmetric DEK is further encrypted by the public key of data owner which is also stored in the cloud and is made available to all the legitimate recipients in accordance with the access control. The cloud maintained here is a public cloud and is a semi-trust as because it does not allows the owner(1)- *recipient two way handshake. This scheme leverages the security by delivering a re-encryption scheme from private key of data owner and public key of recipient by running a proxy service in public cloud environment. We further propose multi-proxy and randomized-proxy scheme to ameliorate the security and robustness of the public cloud.

Keywords: Access control, Cloud Computing, Cloud Storage, Public Key Cryptography, Re-encryption key.

1. INRODUCTION

The most prominent issue that affects cloud performance is security. With data ownership getting isolated with its storage, the control access and usage of shared data remains a point of concern. This confers that data entrust must be highly relied on the recipients rather than on cloud service meaning that stored data should not result to any clear data to the cloud service provider. This is possible through symmetrically encrypting the data before storing the data in cloud itself, enforcing the data security in cloud by not exposing the symmetric key to the cloud . Key management is another tedious issue that we come across. Key management can be a burdensome to the data owner[5]. This arises when the data owner manages the keys himself and DEK(Data Encryption Key) is encrypted using the recipients public key, with the increase in number of trust recipients sharing the data and providing a format that results in decrypting the format with the recipients private key becomes highly undesirable to the data owner[16]. As today data sharing by recipient is getting highly pervasive[4], because of which data owner himself generating re-encryption keys maximize the complexity of cloud performance. Instead, this can be achieved by letting cloud to do the computation ensuring that the cloud remains a semi-trust.

Cloud is a resource full of modeling computations. Since, permitting the data owner to enable the cloud to manage the keys will reduce the cost of computation[5], although there still lies security issues letting cloud to do the computation as cloud is remained as a semi-trust. As cloud remains semi-trust any clear data should not be exposed in cloud. The distribution of key to the users must be decided on the basis of the access control list that data owner decides whom he considers as a trust. This does not allow a two way hand shake between the data owner and the recipient.

To demonstrate these policies a proxy re-encryption scheme is deployed in cloud that is certificateless meaning that the data owner need not to posit his validity as a trust unless he is not a legitimate user of cloud where the data is deployed[17]. In this mechanism the data owner encrypts the data on a symmetric data encryption key that results in obtaining a public key and a private key for the data owner[1,2]. The encrypted data is stored in cloud. On receiving a request for sharing the data from a legitimate user a re-encryption key is generated with the private key of data owner and public key of recipient which is then imputed to an re-encryption algorithm at the proxy server. The re-encryption key and encrypted key of data in cloud contribute to an encrypted format that can be decrypted using the private key of the recipient. Maintaining that cloud is remained as a semi-trust the DEK is never exposed to cloud making the data obfuscated to the cloud.

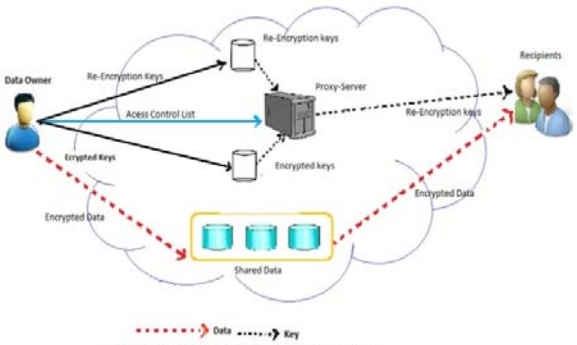
Re-encryption scheme here is a public key cryptography scheme which generates private key and public key to the data owner. This scheme facilitates the recipient to generate his own private and public keys, which reduces the key escrow problem. As the decrypted keys need not to be stored in cloud because recipients have to capability to generate their own decryption keys. The owners also need not prove their authenticity to generate these keys unless they are not a validated users of cloud.

The practicality that this re-encryption scheme is widely employed in public domain where data sharing among the legitimate users is highly evident[3]. One such application can be a social network application[12]. The multiplying size of data in cloud here cannot be estimated[4]. The data can be sharing of files of varied types naming a few like documents, images , software applications, videos. This being a public cloud every recipient can share his own data on the cloud by establishing a link with the re-encryption keys provided by the cloud service provider that is being deployed.

Cloud is persistently exposed to the pernicious environment. It has constant threat to get vulnerable to its data with no time being specified. The proxy if compromised in the malicious environment can manipulate the data and permit access to unauthorized recipients. We resolve the issue by suggesting two possible implementations which would not be able to compromise its proxy key to an adversary. A multi-proxy re-encryption scheme is one such solution where multiple proxy are deployed in several clouds, in such a way that adversary has to either partially or completely compromise with the encrypted key to break down the data. This enhances the security. A randomized proxy re-encryption scheme is another alternative which extends multi proxy-scheme , wherein a random re-encryption key is set out every time a request is made to share the data. This reduces the trust on proxy and further enhancing the security.

2. OVERALL ARCHITECTURE

Cloud maintained here is a semi-trust, that is cloud should not be introduced to any clear data that is deployed in the cloud. This is achieved by encrypting the data with an symmetric data encryption key and is stored in cloud. Thus, even though the encryption key in cloud is known the data still remains obscured as the symmetric DEK is not known to re-encrypt and open the actual data. The data owner need not have to provide the certificates of his authenticity to be a legitimate member of the cloud[7], this proves it as certificateless. The cloud maintains an ACL(access control list) which holds the authenticity of any legitimate recipients who wish to access the data. The ACL is individually maintained for each and every data owner in cloud who wishes to store their data in cloud. A proxy service is maintained which generates an encrypted format that is made available to the legitimate recipient who decrypts it using his own private key. A re-encryption key is generated using the private key of the data owner and the public key of the legitimate recipient. The re-encryption key and the encrypted key of the data stored in the cloud are imputed to re-encryption algorithm present in proxy service that outputs into a format that can be decrypted by the recipient[9].



Proxy Re-encryption scheme for secure data sharing

2.1 Simulation of re-encryption scheme:

The simulation is a two-fold summary in this paper.

1. The data is encrypted by the data owner using a symmetric data encryption key and is stored in cloud. The data owner also places the ACL and the encrypted form of the symmetric DEK with the public key of the data owner in the cloud.

The public key and a private key for the data owner can be generated by using the public key cryptography algorithm like RSA algorithm. The private key is to be kept confidential and public key is exposed to the legitimate users directly for instance like in the social network sites.

2. A recipient when requests to share the data, the proxy service in cloud receives the request and checks the legitimacy of the recipient through the ACL provided by the data owner which is stored in cloud. Then a re-encryption key is generated using the private key of data owner and public key of legitimate recipient. The re-encryption key and the encrypted key of data is imputed to re-encryption algorithm stored in proxy service that can generate a format which is decrypted with the private key of the legitimate recipient[10]. This offers a huge amount of risk when the cloud service provider is exposed to an active attack and the adversary can compromise the ACL and can provide access to the non-legitimate recipients also.

2.2 Properties of re-encryption scheme:

This scheme has several properties that satisfies the security of data sharing with public cloud[11].

1. Unidirectionality: Meaning that the re-encryption of data is not possible to encrypt the data under the recipients public key or decrypt the data under data owners private key. This is unidirectional because recipient must request the data owner to generate the re-encryption key rather than to generate on his own.

2. Non-interactivity: There is no necessity for the recipient to interact with the data owner. Because the data owner can generate the re-encryption key offline by just knowing the recipients public key.

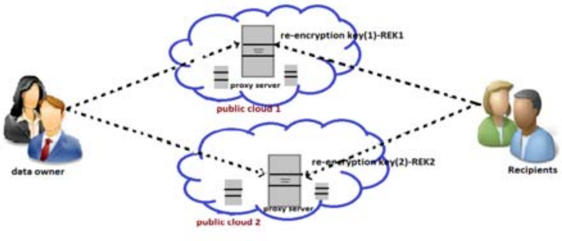
3.Nontransitive: The re-encryption key is to be generated among two parties. If the two parties itself divide into two parts then the proxy can generate its own re-encryption key without the permission of the legitimate users.

4.Single Use: The re-encrypted data should not be further re-encrypted ,if it is possible than the adversary can first re-encrypt the re-encrypted data and then decrypt the data to arrive at the original data.

3. EXTENSIONS FOR RE-ENCRYPTION SCHEME

In this paper we propose two schemes that enhances the security and to reduce the computation problems of re-encryption operation at cloud proxy side[16].

1) Multi Proxy Re-encryption.



Multi-Proxy Re-encryption

To enhance the security of the cloud, we propose multi proxy to support the multiple proxies deployed on different clouds. Like for instance, as we are residing in un-trusted environment like public cloud, which is exposed to many attacks due to misconfiguration of cloud system administrators, or compromised by external attacker with virtualization vulnerability in cloud computing platforms. So, there is a need to protect our data. In multi proxy we store the data not only in one cloud but in many public clouds suppose the Windows azure, gitHub, IBM's blue cloud. If in case the data in one of the cloud is corrupted then the same data available in the other cloud could be accessed . Here the same data in different clouds would generate different proxy re-encryption keys .Hence it is called multi-proxy re-encryption.

Multi proxy re-encryption on the other hand makes the entire system more flexible efficient and scalable[8,13], if same data sharing clouds work properly recipients can choose the proxies to re-encrypt the data according to the physical location[14].

2) Randomized Re-encryption.

Randomized re-encryption enhances the security of multi-proxy by resolving the semi-trust problem. The data owner is facilitated with the usage of re-encryption keys which were previously restricted. If the data owner manages re-encryption keys by himself and does not expose them to the proxy, this will further reduce trust on proxy service. Specifically, after generating a re-encryption key, owner does not send it to proxy. Instead, every time owner wishes to share data to a recipient, he generates a one-time randomized re-encryption key for this session. The one-time key is a randomization of the original re-encryption key, and the cost of generating one-time re-encryption key is lower than that of generating the original re-encryption key[6]. Then owner sends the randomized key along with the data to the proxy. The randomized key can only be used to re-encrypt shared data in the same session.

In randomized proxy re-encryption , proxy does not possess any long term re-encryption keys in the proxy instead, it possesses sessional re-encryption key The proxy cannot complete a re-encryption of any new message with previous re-encryption keys. The data owner randomizes n-different re-encryption keys for n-different recipients. Some methods can be applied to reduce the computation cost for the data owner.

4. CONCLUSION

This paper proposes a re-encryption scheme for secure data sharing with public cloud. Re-encryption scheme in cloud makes the data in cloud secure , flexible and a robust system. The proxy re-encryption is certificateless as the owner and recipient need not authenticate themselves through providing certificates ,the owner and recipient are validated to the cloud that they opt to choose. To enhance the security of cloud the proxy re-encryption is further manipulated into two proposed schemes, a multi proxy re-encryption scheme which manages to run the same data on different cloud service providers with different proxy services. Randomized re-encryption is implemented to reduce the trust of proxy which further implements multi-proxy re-encryption scheme, although the cost of computation increases , data owner takes the burden of generating random sessional re-encryption keys. Proxy re-encryption scheme for secure data sharing is highly advantageous in social network services.

5. REFERENCES

[1] Loknath S, Shivamurthy S, Bhaskar S and

Shantgouda S,"Strong and Secure Re-Encryption Technique to Protect Data Access by Revoked Users in Cloud," International Conference on Advances in Computer and Electrical Engineering (ICACEE'2012) Nov. 17-18, 2012 Manila (Philippines)

[2] Piotr K. Tysowski and M. Anwarul Hasan, "Towards Secure Communication for Highly Scalable Mobile Applications in Cloud Computing Systems," National Sciences and Engineering Research Council (NSERC)

[3] Yan Li, Nakul Sanjay Dhotre, Yasuhiro Ohara, Thomas M. Kroeger, Ethan L. Miller and Darrell D. E. Long, "Horus: Fine-GrainedEncryption-BasedSecurityforLarge-ScaleStorage," 11th USENIX Conference on File and Storage Technologies (FAST ’13)

[4] Sun-Ho Lee and Im-Yeong Lee, "A Secure Index Management Scheme for Providing Data Sharing in Cloud Storage," J Inf Process Syst, Vol.9, No.2, June 2013

[5] Stiffy Sunny and L. Agilandeeswari, "Secure Data Sharing of Patient Record in Cloud Environment using Attribute Based Encryption ," International Journal of Applied Engineering Research, ISSN 0973-4562, Vol. 8, No. 19 (2013) © Research India Publications

[6] Raluca Ada Popa, Jacob R. Lorch, David Molnar, Helen J. Wang and Li Zhuang, "Enabling Security in Cloud Storage SLAs with CloudProof "

[7] Satyendra Singh Rawat and Mr. Alpesh Soni, "A Survey of Various Techniques to Secure Cloud Storage," National Conference on Security Issues in Network Technologies (NCSI-2012) August 11-12,2012

[8] J. Raghavi and S. Krishna Anand, "A Survey on Cloud Storage Systems and Encryption Schemes," International Journal of Engineering and Technology (IJET)

[9] Aurelia Delfosse, Jeremy Fanton, Thierry Floriani, Vincent Malguy, Nargisse Marine and Cedric Tavernier, "Cloud Data security and privacy in IAAS model," Recent Advances in Computer Science and Networking

[10] Pei-Shan Chung1, Chi-Wei Liu2 and Min-Shiang Hwang, "A Study of Attribute-based Proxy Re-encryption Scheme in Cloud Environments," International Journal of Network Security, Vol.16, No.1, PP.1-13, Jan. 2014

[11] Keying Li, Jianfeng Wang, Yinghui Zhang and Hua Ma, "Key Policy Attribute-based Proxy Re-encryption and RCCA Secure Scheme, "Journal of Internet Services and Information Security (JISIS), volume: 4, number: 2, pp. 70-82

[12] Vivek Shandilya and Sajjan Shiva, "Security in the Cloud: A stake holder’s perspective "

[13] P. R. Jaiswal and A. W. Rohankar, "Attribute-Based Encryption: An Efficient Way to Secure Cloud Storage," International Journal of Scientific & Engineering Research, Volume 4, Issue 11, November-2013

[14] Mrs. Pooja A. Uplenchwar and Mrs. L. H. Patil, "Data Security in Unreliable Cloud Using Access Control and Access Time," International Journal of Scientific & Engineering Research, Volume 4, Issue 12, December-2013

[15] Nicholaus Gati and K. Suresh Babu, "Enforcing Access Control Delegation to Secure and Preserve Privacy of Cloud Data," International Journal of Scientific & Engineering Research, Volume 5, Issue 8,August-2014

[16] Lei Xu, Xiaoxin Wu and Xinwen Zhang, "CL-PRE: a Certificateless Proxy Re-Encryption Scheme for Secure Data Sharing with Public Cloud," ASIACCS ’12, May 2–4, 2012, Seoul, Korea. Copyright 2012 ACM 978-1-4503-1303-2/12/05

[17] Chittaranjan Hota, Sunil Sanka, Muttukrishnan Rajarajan and Srijith K. Nair, "Capability-based Cryptographic Data Access Control in Cloud Computing," Int. J. Advanced Networking and Applications Volume: 01 Issue: 01 Pages: (2011)