Multi-Cloud Key Aggregation for Scalable Data Sharing

Mr. G. R. Shinde[1], Prof. K. N. Shedge[2]

[1]PG-Student, Department of Comp. Engg., SVIT COE, Sinnar, Maharashtra, India
[2]Assistant Professor, Department Comp. Engg, SVIT COE, Sinner, Maharashtra, India

***Abstract*—*In cloud computing the process of sharing data must be secure for the security reason. The proposed system provides two phase security mechanism. In that encryption data is first phase and splitting is another phase. It provides securely, efficiently, and flexibly shares data with others in multi cloud storage using aggregate cryptosystem concept. It describes new public-key cryptosystems that produce constant-size cipher texts such that efficient delegation of decryption rights for any set of ciphertext is possible that decrypted files are spited and stored on the different clouds for the security reason. The process of splitting and merging can be carried out by the Shamir secrets algorithm. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but encompassing the power of all the keys being aggregated. In other words, the secret key holder can release a constant-size aggregate key for flexible choices of ciphertext set in cloud storage, but the other encrypted files outside the set remain confidential. The secure aggregate key sent to other users or be stored in a smart card with very limited secure storage. It also provides formal security analysis of our schemes in the standard model.*

Keywords- Key Aggregation, Multi-Cloud Storage, Authentication, Scalable Data Sharing.

I. INTRODUCTION

The most important aspect of cloud is security. Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25 GB. Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication which means any unexpected privilege escalation will expose all data. In a shared-tenancy cloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM co resident with the target. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owner’s anonymity. Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, for example, with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These users are motivated to encrypt their data with their own keys before uploading them to the server.

Data sharing is an important functionality in cloud storage. For example, bloggers can let their friends view a subset of their private pictures; an enterprise may grant her employees access to a portion of sensitive data. The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial. Encryption keys also come with two flavors—symmetric key or asymmetric (public) key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she or he has to give the secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are different in public key encryption. The use of public-key encryption gives more flexibility for our applications. For example, in enterprise settings, every employee can upload encrypted data on the cloud storage server without the knowledge of the company’s master-secret key.

II. BACKGROUND(LITERATURE REVIEW)

As per the survey of the existing systems following points are observed:

1. Wen-GueyTzeng proposed a time-bound cryptographic key assignment scheme in which the cryptographic keys of a class were different for each time period, that was the cryptographic key of class Ci at time r is K(i,t).Key derivation is constrained not only by the class relation, but also the time period. In our scheme, each user holds some secret parameters whose number is independent of the number of the classes in the hierarchy and the total time periods. We present two novel applications of our scheme. One is to broadcast data to authorized users in a multilevel security way and the other is to construct a flexible cryptographic key backup system.

2. Cong Wang proposed a secure cloud storage system supporting privacy-preserving public auditing. These techniques extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

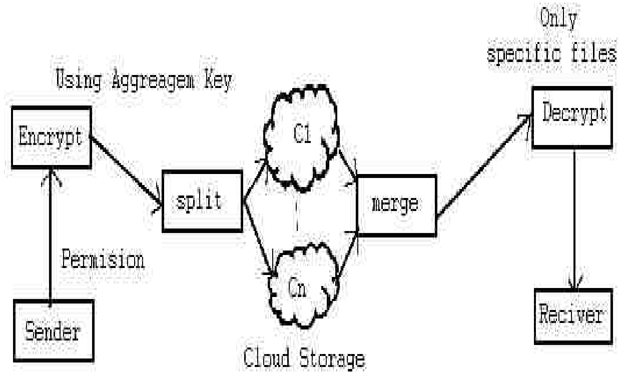
3. Xiaoming Huet. al. proposed a Gentry's identity-based encryption scheme, we give a construction for an ID-PRE scheme that is fully secure in the standard model. Proposed scheme has the following advantages comparison with all previous ID PRE Schemes: Short Public Parameters, a tight reduction and fully security in standard model.

4. Laurence T. Yang et al. propose a scalable two-phase top-down specialization (TDS) approach to anonymize large-scale data sets using the MapReduce framework on cloud. In both phases of our approach, we deliberately design a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. Experimental evaluation results demonstrate that with our approach, the scalability and efficiency of TDS can be significantly improved over existing approaches.

5. Surya Nepal proposed a secure cloud storage service architecture with the focus on Data Integrity as a Service (DIaaS) based on the principles of Service Oriented Architecture and Web services. Our approach not only releases the burdens of data integrity management from a storage service by handling it through an independent third party data Integrity Management Service (IMS), but also reduces the security risk of the data stored in the storage services by checking the data integrity with the help of IMS. We define data integrity protocols for a number of different scenarios, and demonstrate the feasibility of the proposed architecture, service and protocols by implementing them on a public cloud, Amazon S3. We also study the impact of our proposed protocols on the performance of the storage service and show that the benefits of our approach outweigh the little penalty on the storage service performance.

III. PROPOSED SYSTEM

A key-aggregate encryption scheme consists of five polynomial-time algorithms. The data owner establishes the public system parameter via Setup and generates a public/master-secret3 key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract. The generated keys can be passed to delegates securely (via secure e-mails or secure devices) Finally, any user with an aggregate key can decrypt any ciphertext provided that the cipher text's class is contained in the aggregate key via Decrypt.



.

Fig. 1. Architecture of Proposed System

1. Setup(1, n): executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1_ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter pram, which is omitted from the input of the other algorithms for brevity.

2. Permission(): It selects the appropriate files for the specific users. It is one type of access right module.

3. Add Circle(): It is on type of group. It is used to send a data to the specific users. It saves the time of user to select each user individually. Users have full authority to create its own separate groups or circles to save time and some effort.

4. KeyGen(pk, msk): executed by the data owner to randomly generate a public/master-secret key pair (pk; msk).

5. Encrypt(pk, i, m): executed by anyone who wants to encrypt data. On input a public-key pk, an index i denoting the ciphertext class, and a message m, it outputs a ciphertext C.

6. Split() and Merge(): It is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part, where some of the parts or all of them are needed in order to reconstruct the secret.

7. Extract(msk, S): executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegate. On input the master secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by KS.

8. Decrypt(KS, s, i, C): executed by a delegate who received an aggregate key KS generated by Extract. On input KS, the set S, an index i denoting the ciphertext class the ciphertext C belongs to, and C, it outputs the decrypted result m if i £ S. User only able to decrypt those files which are accessible to that user.

Example of Shamir Secretes Algorithm

Suppose that our secret is 1234 $(S = 1234)$.
We wish to divide the secret into 6 parts (n=6), where any subset of 3 parts (k=3) is sufficient to reconstruct the secret. At random we obtain two (k-1) numbers: 166 and 94.

$$\mathbf{(a_0 = 1234;\ a_1 = 166;\ a_2 = 94)}$$

Our polynomial to produce secret shares (points) is therefore: **f(x)= 1234 + 166x +94x^2**

We construct 6 points D_{x-1} =(x,f(x)) from the polynomial:
D_0 = (1,1494); D_1 = (2,1942); D_3 = (32578);
D_4 = (43402); D_5 = (54414); D_6 = (65614)

We give each participant a different single point (both x and f(x)). Because we use D_{x-1} instead of D_x the points start from (1, f(1)) and not (0,f(0)). This is necessary because if one would have(0,f(0)) he would also know the secret (S=f(0)).

IV. RECONSTRUCTION OF DATA

This section In order to reconstruct the secret any 3 points will be enough.

Let us consider,

$$(x_0, y_0) = (2, 1942); (x_1, y_1) = (4, 3402); (x_2, y_2) = (5, 4414).$$

We will compute Lagrange basis polynomials:

$$\ell_0 = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} = \frac{x-4}{2-4} \cdot \frac{x-5}{2-5} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}$$

$$\ell_1 = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} = \frac{x-2}{4-2} \cdot \frac{x-5}{4-5} = -\frac{1}{2}x^2 + \frac{7}{2}x - 5$$

$$\ell_2 = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} = \frac{x-2}{5-2} \cdot \frac{x-4}{5-4} = \frac{1}{3}x^2 - 2x + \frac{8}{3}$$

Therefore,

$$f(x) = \sum_{j=0}^{2} y_j \cdot \ell_j(x)$$

=1234 + 166x +94x^2

Recall that the secret is the free coefficient, which means that **S = 1234**, and we are done.

V. RESULT

Figure 2 shows the snapshot of splitting process of encrypted file. Here file is splitted in 6 parts. User can directly upload the splitted part on to the clouds. We are providing the facility of multi cloud so Google Drive and Dropbox both options are given.

If user wants to see the splitted file then he/ she can click on the retrieve link. Splitted file contains the only numbers i.e. cipher text.

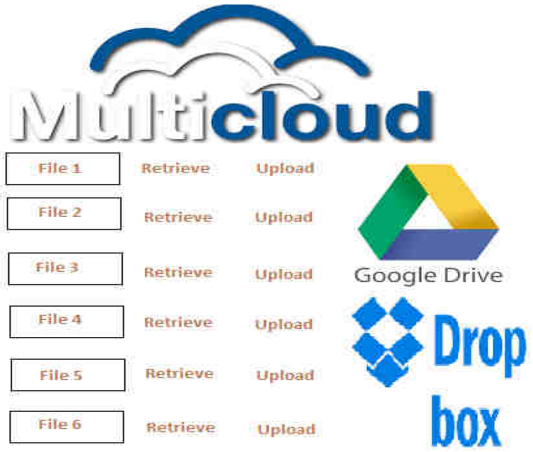


Fig. 2. Splitting Result of Module 3

VI. CONCLUSION

In this paper we finally conclude that our proposed system very secure because it provides two phase security first one: Encryption with strong aggregate key second: Splitting encrypted file and storing on different cloud. Currently we have used two clouds i.e. Google Drive and Drop Box but in future we will use more and different clouds to provide more security.

REFERENCES

[1] Cheng-Kang Chu, Sherman S.M. Chow, Wen-GueyTzeng, Jianying Zhou, Robert H. DengIEEE, “Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage.”IEEE Trans. parallel and distributed systems, VOL. 25, NO. 2, FEBRUARY 2014.

[2] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy- Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[3] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, “Dynamic and Efficient Key Management

for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.

[4] R.S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2, pp. 95-98, 1988.

[5] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.

[6] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.

[7] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M.Yiu, "SPICE – Simple Privacy-Preserving Identity-Management for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

[8] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

[10] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.

[11] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology (CRYPTO '01), vol. 2139, pp. 213-229, 2001.

[12] Y. Sun and K.J.R. Liu, "Scalable Hierarchical Access Control in Secure Group Communications," Proc. IEEE INFOCOM '04, 2004.

[13] S.S.M. Chow, C.-K.Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.

[14] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '05), vol. 3494, pp. 457-473, 2005.

[15] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM J. Computing,vol. 36, no. 5, pp. 1301-1328, 2007.

[16] T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu, "Identity-Based Encryption Resilient to Continual Auxiliary Leakage," Proc.Advances in Cryptology Conf. (EUROCRYPT '12), vol. 7237, pp. 117-134, 2012.

[17] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," Proc. 14th AustralasianConf.Information Security and Privacy (ACISP '09), vol. 5594,pp. 327-342, 2009.

[18] L. Hardesty, Secure Computers Aren't so Secure. MIT press, http:// www.physorg.com/news176107396.html, 2009.