A Cryptographic Solution for Secured Data Sharing in Cloud Storage using Single Master-Key Technique

Archana BR[1] Dr. Siddaraju[2]
[1]M.Tech Student [2]Head of the Department
[2]Department of Computer Science & Engineering
[1,2]Dr.AIT, Bangalore, KA-India

Abstract— This paper propose a new idea to securely share the data in cloud storage. The paper explores a new Public Key Cryptosystem, which uses Single Master-Key Technique to delegate the data/files from one user(say sender A) to the other user (say receiver B).This cryptosystem produces a fixed-size (comparatively small) ciphertexts such that, the rights of decrypting any set of ciphertexts can be transferred efficiently in cloud. This could be possible by generating a concise key-we call it as single-master key- which encompasses the power of all the encryption key of the set of data/files to be transferred. The other files/data will be kept secure and confidential. This concise key can be sent to others or can be stored in a smart card and it requires very limited and secure storage.

Key words: Public Key Cryptosystem, Ciphertext, Cloud Storage, Concise-Key

I. INTRODUCTION

Now a days Cloud storage is very popular. In organizations and industries, we can see data outsourcing is on very high demand, which assists in the strategic management of corporate data. In our daily life, it is easy to apply for free accounts like email, photo album, file transferring and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). It is also used as a core technology behind many online services for personal applications. 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.

One of the important functionality of cloud storage is Data sharing between the end parties. The one of the most challenging problem in data sharing is how to effectively share encrypted data/files. Sender can download the encrypted data from the cloud storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage and encrypting the files, storing them is of no use. Users should be able to transfer the access rights of the sharing data to others so that they can access these data from the server directly.

A. A Scenario of Drop-Box:

Assume that Ramesh puts all his private photos on Drop box, and he does not want to expose his photos to everyone. Due to various data leakage possibility Ramesh cannot feel relieved by just relying on the privacy protection mechanisms provided by Drop-box, so he encrypts all the photos using his own secret keys before uploading. One day, Ramesh's friend, Dinesh, asks him to share the photos taken over all these years which Dinesh appeared in. Ramesh can then use the share function of Drop-box, but the problem now is how to delegate the decryption rights for these photos to Dinesh.

A possible option Ramesh can choose is to securely send Dinesh the secret keys involved. There are two ways of transferring the files with delegation rights, i.e secret keys as follows.

1) Ramesh encrypts all files with a single encryption key and gives Dinesh the corresponding secret key directly.
2) Ramesh encrypts files with distinct keys and sends Dinesh the corresponding secret keys.

The first method, is not good enough since all unchosen data may be also leaked to Dinesh. In the second method, the number of such keys is as many as the number of the shared photos. If say, a thousand number of such files to be transferred then transferring these secret keys inherently requires a secure channel, and storing these keys requires an expensive secure storage. The costs and complexities involved in this case increase with the number of the decryption keys to be shared. In short, it is very heavy and costly to do that. The system architecture which describes the proposed system is as shown in the figure (Fig. 1a)

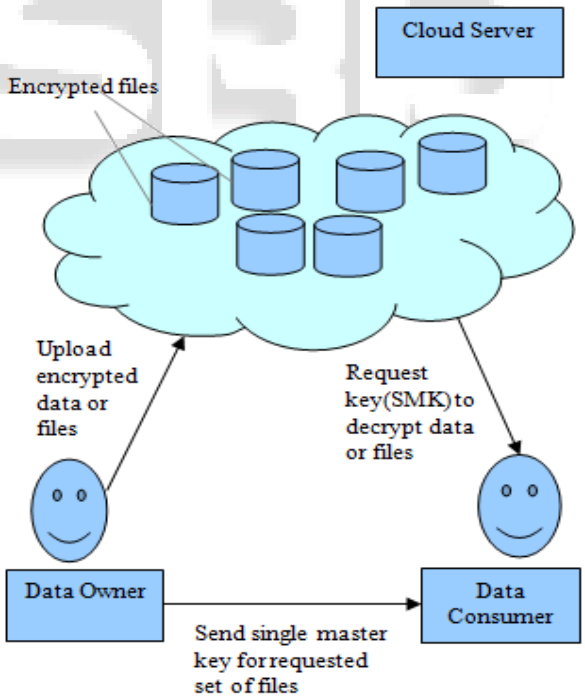

Fig 1a: System architecture of our SMK Technique.

Therefore, the best solution for the above problem is that Ramesh encrypts files with distinct public-keys, but only sends Dinesh a single (constant-size) decryption key or a single Master-Key. With our solution Ramesh can simply send Dinesh a Single Master-Key via a secured E-mail. Dinesh can download the encrypted photo from Ramesh's Drop-box space and then use this key to decrypt those encrypted photos.

This project proposes a new Public Key Cryptosystem in which a single master-key is generated for a particular set of encrypted data. In other words, the secret key holder can release a concise key called Single Master-Key for flexible choices of cipher text set in cloud storage. This compact key can be sent to others or be stored in a smart card with very limited secure storage.

II. RELATED STUDIES

There are many cryptographic solutions are exists which are based on type of encryption that is either symmetric key encryption or public key encryption. In this section we would discuss about some of the encryption schemes focusing on their advantages and disadvantages.

A. Cryptographic Solution for Hierarchical Storage Systems

Under this section we are discussing about many general Cryptographic or security schemes which are in use to minimize the cost in storing and in managing the secret key for general tree based systems.In hierarchical storage systems(5)(14), the key size is not constant. It may vary depends upon the tree structure.Using this tree structure we can derive the keys for the decendent nodes by knowing the key of the parent node. Most of these schemes produce keys using symmetric key cryptosystems. The furthur key derivations for child nodes requires modular arithmetic or public key cryptosystem.Hierarchical approaches can solve the problems partially. If the owner wants to share all the files under certain branch then hierarchical approach is useful.

B. Symmetric Key Encryption

Symmertic key encyption (16) is proposed for transmitting large number of data using a compact key in broadcast manner. This scheme acheives the same properties as in our prooposed scheme.It is designed for symmetric keys settings.This scheme tries to reducce the key size for achieving authentication in symmetric key encryption(16).

C. Identity Based Encryption(IBE).

Identity based encryption is a type of public key encryption (6)(13)(15) uses random orales to generate a aggregate key. In this scheme the keys to be aggregate must come from different identity division.Here the key aggregation comes at an expense of O(n) size for both ciphertexts and the public system parameter, where n is the number of secret keys.Cost of storing iphertext is expensive in this scheme.

D. Attribute Based Encryption.

This scheme (7) allows each ciphertext to be allocated with an attribute.the master secret key holder can extract the secret key for a policy of these attributes so the ciphertext can be decrypted using this key. ABE is collusion resistance but the size of key will increases linearly with number of attributes. Also the ciphertext size is also not constant.

E. Proxy Re Encryption (PRE) Scheme

This is the method where the decryption power will be transferred to the receiver without sending the secret key. PRE (8)(10)(11).Proxy only converts the users encrypted files or ciphertexts. The user should have full trust on the proxy itself. The main disadvantage of PRE proxy should not reside on the server also every decryption requires separate interaction with the proxy.

The table summarises the above discussed encryption methods.

No.	Cryptographic Schemes	Key Size	Size of the Cipher text	Type (Encryption)
1	Cryptographic solution for hierarchical storage systems.	Depends on tree structure (non constant.).	constant	SymmetricThe table summarises the above discussed encryption methods.
2	Symmetric key encryption	constant	constant	Symmetric key
3	Identity Based encryption (IBE)	constant	nonconstant	Public key
4	Attribute based encryption(ABE)	Non constant	Non constant	Public key
5	Proxy Re Encryption(PRE	NA	NA	Public key

Table 1: Summary of Different Encryption Methods.

III. THE DESIGN OF SYSTEM STRUCTURE

The modular design of this system is explained this section. The basics of broadcast encryption scheme (8) is used as the inspiration in this design. We are mainly concentrating on a constant sized concise-key and to achieve this we are providing five polynomial time algorithms as follows. Here we are using three main modules,

- The data owner generates a public system parameter through a PP-Setup Algorithm.
- A public key and single-master key pair is also generated by data owner via executing a Key-Generation Algorithm.
- An Encrypt Algorithm can be executed by user or data owner before uploading. The cipher text for the file or data will be produced by this algorithm using the public key.
- Again data owner will be responsible for execute the SMK-Extract Algorithm and it will produce the single master key which is used by the user who wants to download the encrypted files on request. Here for a particular set (S) of cipher texts a single master key will be produced.
- Finally a Data Decrypt algorithm is used to decrypt the cipher text when the user gets the single-master key. On getting the single-master key for set S and an index (i)which indicates the class of cipher text we can decrypt the files.

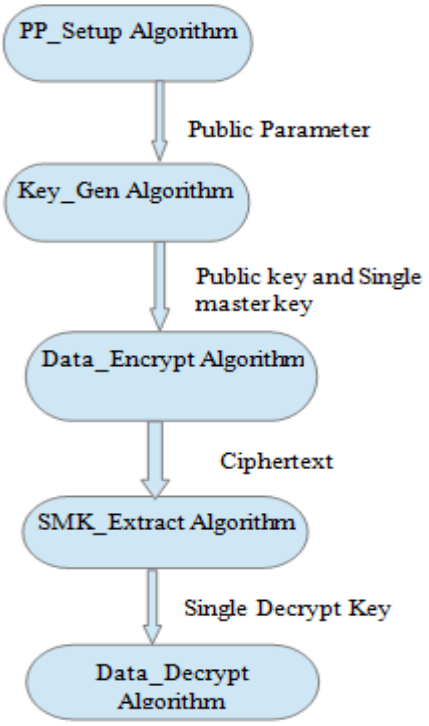

Fig. 3a: Flow of the Five Polynomial Time Algorithms

The construction of SMK technique uses Bilinear mapping or multilinear mapping.Let G and G_m be two cyclic groups of order p (prime number) and the mapping between them m : G×G→ G_m. The Bilinear property for the mapping of two groups can be defined as,

$\forall g_1,g_2 \in G$, x,y $\in Z$, a Integer set, $m(g_1^x.g_2^y)=m(g_1.g_2)^{xy}$

Also the mapping $m(g_1,g_2) \neq 1$(Nondegenerate property).

Thus, by using these two properties the five algorithms are goes like this.

PP-Setup(1^λ, n):

Input: Randomly select a bilinear cyclic group G with order p.

Pick an object g $\in$ G and l $\in Z_p$., set of non-zero prime integer.

Compute: $g_i=g^l \in G$ for i={1,....,n,n+2,.....2n}.

Output: Public parameter pp={g_1,.....g_n, g_{n+2},.....g_{2n}}=g_i.

Key-Generation:

Input: Randomly pick an integer r $\in Z_p$.

Output: Public Key pk = g^r

Single-master key smk = r.

Data Encrypt:

Input: F file belonging to G_T and i $\in$ {1,2,3,.....,n}

Select t $\in$ Z_p randomly.

Output: Ciphertext C={g^t * $(g^r g_i)^t$,F*m(g_1 .g_2)^t}.

SMK_Extract(smk, S):

Input: Smk obtained from Key-Generation Algorithm and

A set of files S with index j,

S_j ={ f_1 ,f_2 , …. , f_n } and j={1,2,.....,n}

Compute: Single Decrypt Key (Sdk) as follows.

$$Sdk = \Sigma_{j \in S} g^r{}_{n+1-j}$$

Output: Pass the Sdk to the Data-Decrypt Algorithm upon user request.

Data_Decrypt(Sdk,S,i,C):

Input: Sdk- Single Decrypt Key produced by SMK_Extract Algorithm.

S the set of files.

Set of ciphertext with index i

Compute: If i $\in$ S

Compute F : F= C_3 * m (Sdk * $\Sigma_{j \in S} g_{n+1-j+i}$, C_1) / m ($\Sigma_{j \in S} g_{n+1-j}$, C_2)

Return files F.

Otherwise

Return Error.

Using this SMK technique the user A can transfer the decryption rights to user B so that he/she can download the files on request more effectively, efficiently and also securely.

We implemented this SMK technique using java with JPBC (Java Pairing Based Cryptography) Library[6]. The implementation supports multithreading and uses memory mapped files to save in primary memory requirements. We have used Type A bilinear mapping to generate the public system parameter. This Type A pairing are constructed on the curve $y^2=x^3+x$ .For Type A pairing, JPBC provides a ported and a PBC wrapped generator and it is symmetric pairing.

IV. Conclusion

As cloud storage is gaining imporatance, we need to protect the data privacy. This becomes a challenging question of cloud storage. To achieve this we are proposing a new cryptographic solution with a compressed secret key for a class of ciphertext in cloud storage. The data consumer have the decryption rights by SMK technique which generates the concise key called single master key ,using which one can decrypt the required set of files. Main advantage is the size of secret key is constant . The only limitation is the ciphertext size is also constant.It is better solution if there is no dependncy on ciphertext classes.

Reference

[1] L. Hardesty, “Secure computers aren’t so secure,” MIT press, 2009,http://www.physorg.com/news176107396.html.
[2] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy- Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans.
[3] Computers, vol. 62, no. 2, pp. 362–375, 2013.
[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, “Storing Shared Data on the Cloud via Security-Mediator,” in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.
[5] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” in Proceedings of Advances in Cryptology - EUROCRYPT ’03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.
[6] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, “Dynamic and Efficient Key Management for Access Hierarchies,” ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.
[7] F. Guo, Y. Mu, and Z. Chen, “Identity-Based Encryption: How to Decrypt Multiple Ciphertexts

Using a Single Decryption Key," in Proceedings of Pairing-Based Cryptography (Pairing '07), ser.LNCS, vol. 4575. Springer, 2007, pp. 392–406.
[8] M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," in ACM Conference on Computer and Communications Security, 2009, pp. 121–130.
[9] R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07). ACM, 2007, pp. 185–194.
[10] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption Without Random Oracles," in Information Security Conference (ISC '07), ser. LNCS, vol. 4779. Springer, 2007, pp. 189–202.
[11] C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re-Encryption," in Australasian Conference on Information Security and Privacy (ACISP '09), ser. LNCS, vol. 5594. Springer, 2009, pp. 327–342.
[12] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 1, pp. 1–30, 2006.
[13] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," in Proceedings of Advances in Cryptology - CRYPTO '05, ser. LNCS, vol. 3621. Springer, 2005, pp. 258–275.
[14] T. H. Yuen, S. S. M. Chow, Y. Zhang, and S. M. Yiu, "Identity- Based Encryption Resilient to Continual Auxiliary Leakage," in Proceedings of Advances in Cryptology - EUROCRYPT '12, ser. LNCS, vol. 7237, 2012, pp. 117–134.
[15] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494. Springer, 2005, pp. 440–456.
[16] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM Journal on Computing (SIAMCOMP), vol. 36, no. 5, pp. 1301–1328, 2007.
[17] http://en.wikipedia.org/wiki/Symmetric-key_algorithm

