IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

A SECURE WAY FOR KEY DISTRIBUTION WITHOUT ANY SECURE COMMUNICATION

LAKKIMSETTI SHANMUKHA V N DHANA RAJU[1], SAIPRIYA VISSAPRAGADA[2]
[1]PG Scholar, Dept of CSE, Srinivasa Institute of Engineering & Technology, Cheyyeru, Amalapuram-A.P, India
[2]Associate Professor, Dept of CSE, Srinivasa Institute of Engineering & Technology, Amalapuram-A.P, India

Abstract- Benefited from cloud computing, users can achieve an effective and economical approach for data sharing among group members in the cloud with the characters of low maintenance and little management cost. Meanwhile, we must provide security guarantees for the sharing data files since they are outsourced. Unfortunately, because of the frequent change of the membership, sharing data while providing privacy-preserving is still a challenging issue, especially for an untreated cloud due to the collusion attack. Moreover, for existing schemes, the security of key distribution is based on the secure communication channel, however, to have such channel is a strong assumption and is difficult for practice. In this paper, we propose a secure data sharing scheme for dynamic members. Firstly, we propose a secure way for key distribution without any secure communication channels, and the users can securely obtain their private keys from group manager. Secondly, our scheme can achieve fine-grained access control, any user in the group can use the source in the cloud and revoked users cannot access the cloud again after they are revoked. Thirdly, we can protect the scheme from collusion attack, which means that revoked users cannot get the original data file even if they conspire with the untreated cloud. In our approach, by leveraging polynomial function, we can achieve a secure user revocation scheme. Finally, our scheme can achieve fine efficiency, which means previous users need not to update their private keys for the situation either a new user joins in the group or a user is revoked from the group.

1. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

Fig: FigStructure of cloud computing

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games. The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computation.

2. SYSTEM MODEL AND ASSUMPTIONS

2.1 Threat Model

As the threat model, in this paper, we propose our scheme based the only way to protect the information from attacking by the passive eavesdroppers and active saboteurs is to design the effective security protocols. This means there are

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

not any secure communication channels between the communication entities. Therefore, this kind of threaten model can be more effective and practical to demonstrate the communication in the real world.

2.2System Model

Group members (users) are a set of registered users that will store their own data into the cloud and share them with others. In the scheme, the group membership is dynamically changed, due to the new user registration and user revocation.

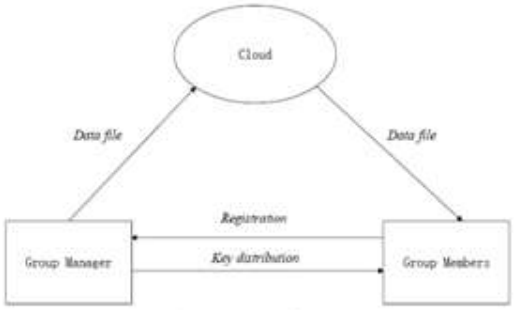


Figure 1 System model

2.3 Design Goals: We describe the main design goals of the proposed scheme including key distribution, data confidentiality, and access control and efficiency as follows: Key Distribution: The requirement of key distribution is that users can securely obtain their private keys from the group manager without any Certificate Authorities. In other existing schemes, this goal is achieved by assuming that the communication channel is secure, however, in our scheme, we can achieve it without this strong assumption. Access control: First, group members are able to use the cloud resource for data storage and data sharing. Second, unauthorized users cannot access the cloud resource at any time, and revoked users will be incapable of using the cloud resource again once they are revoked. Data confidentiality: Data confidentiality requires that unauthorized users including the cloud are incapable of learning the content of the stored data. To maintain the availability of data confidentiality for dynamic groups is still an important and challenging issue. Specifically, revoked users are unable to decrypt the stored data file after the revocation. Efficiency: Any group member can store and share data files with others in the group by the cloud. User revocation can be achieved without involving the others, which means that the remaining users do not need to update their private keys.

3 THE PROPOSED SCHEME

3.1.1 Preliminaries

3.1.1 Bilinear Maps

Let 1 *G* and 2 *G* be additive cyclic groups of the same prime order q [18]. Let

e :*G*1 □*G*1 □*G*2 denote a bilinear map constructed with the following properties:

1. Bilinear: For all *, *q a b*□*Z* and □□□□□□□□□1 , , , , *.ab*

P Q□*G e AP bQ* □□*e P Q*

2. No degenerate: There exists a point Q such that *e*□*Q*, *Q*□□□1.

3. Computable: There is an efficient algorithm to compute *e*□*P*, *Q*□□for any1 *P*, *Q*□*G*.

3.1.2 Complexity Assumptions

Definition 1

(Basic Diffie-Hellman Problem (BDHP) Assumption. Given base

Point *P* and a value * , *q* □□□*Z* it is easy to compute □□□*P*. However, given *P*,□□□*P*, it is infeasible to compute □□because of the discrete logarithm problem.

Definition 2

(Decisional Diffie-Hellman Problem (DDHP) Assumption [20]). Similar

to definition 1, given base point *P* and *aP*,□*a* □*b*□*P*, it is infeasible to compute *b*P.

Definition 3

For unknown * , *q a*□*Z* given 21 , , ,..., , , *l Y aY a Y a Y P*□*G* it is infeasible to compute□□□1 *e Y*,*P a* .

3.1.3 Notations

Each user has a pair of keys *pk*, SK which is used in the asymmetric encryption algorithm, and *pk* needs to be negotiated with the group manager on the condition that no Certificate Authorities and security channels are involved in. *KEY* is the private key of the user and is used for data sharing in the scheme. *UL* is the group user list which records part of the private keys of the legal group users. *DL* is the data list which records the identity of the sharing data and the time that they are updated.

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

Notation	Description
ID_i	the identity of user i
ID_{data_i}	the identity of data $_i$
pk	the public key of the user that needs to be negotiated with the group manager
sk	the corresponding private key to pk
$KEY=(x,A,B)$	the private key which is distributed to the user from the group manager and used for data sharing
$Enc_k(\ )$	symmetric encryption algorithm used the encryption key k
$AENC_k(\ )$	asymmetric encryption algorithm used the encryption key k
UL	group user list
DL	data list

Figure 2 Notation used in our scheme description

3.1.4 File Upload

The operation of file upload is performed as illustrated in Figure. First of all, the group member chooses a unique data file identity and a random number then computes these parameters as the following equation: data ID *, q k Z □

$$\begin{cases} C_1 = k \cdot Y \in G_1 \\ C_2 = k \cdot P \in G_1 \\ K = Z^k \in G_2 \\ C = Enc_K(M) \end{cases} \quad (2)$$

Then the group member encrypts $(ID_{data_i}, C_1, C_2, C, t_{data_i})$ with his private key g_i, where t_{data_i} is the real time stamp. At last, the group member sends $Enc_{g_i}(ID_{data_i}, C_1, C_2, C, t_{data_i})$ to the group manager.

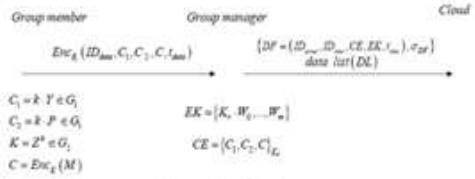


Figure 4 File upload

After getting this message the group manager decrypts it and gets then the group manager checks the legal group members in his local storage space and if is the private key of a legal user, then the group manager constructs the polynomial function and the exponential function After that, the group manager selects a random re-encryption key and constructs Finally, the group manager encrypts cipher-text with the re-encryption key and sends to the cloud, where is the time that the data file is uploaded and is the signature of the group manager for the data file

In addition, the group manager also sends the data list to the cloud in order to let the usersverify the freshness of the data file.As illustrated in Table 2, the group manager adds and the current time to the data list DL. To guarantee that both users can obtain the latest version of data file and the cloud can updatethe data file, the group manager updates the data list everyday. At last, the group manager data data ID t DL t

Table 2 Data list

ID_{group}	ID_{data_i}	t_{data_i}	
	ID_{data_1}	t_{data_1}	
	•	•	
	ID_{data_n}	t_{data_n}	t_{DL} $sig(DL)$

Finally, on receiving the message, the cloud verifies the identity of the group manager by checking the equation $e(W', f_1(DF')) = e(P, \sigma_{DF})$ and stores the message after successful verification.

3.2.4 User Revocation

User revocation is performed by the group manager and the cloud, which is illustrated in figure 5.

Figure 5 User revocation

3.2 Overview and Rationale

Generating unique differentiable copies of the data file is the core to design a provable multi-copy data possession scheme. Identical copies enable the CSP to simply deceive the owner by storing only one copy and pretending that it stores multiple copies. Using a simple yet *efficient* way, the proposed scheme generates distinct copies utilizing the *diffusion* property of any secure encryption scheme. The diffusion property ensures that the output bits of the ciphertext depend on the input bits of the plaintext in a very complex way, *i.e.*, there will be an unpredictable complete change in the ciphertext, if there is a single bit change in the plaintext. The interaction between the authorized users and the CSP is considered through this methodology of generating distinct copies, where the former can decrypt/access a file copy received from the CSP. In the proposed scheme, the authorized users need only to keep a single secret key (shared with the data owner) to decrypt the file copy, and it is not necessarily to recognize the index of the received copy. In this work, we propose a MB-PMDDP scheme allowing the data owner to update and scale the blocks of file copies outsourced to cloud servers which may be untrusted. Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner. Moreover, the verifier should be aware of the block indices to guarantee that the CSP has inserted or added the blocks at the requested positions in all copies. To this end, the

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

proposed scheme is based on using a small data structure which we call a map-version table.

3.3 Map-Version Table

The map-version table (MVT) is a small *dynamic* data structure stored on the verifier side to validate the integrity and consistency of all file copies outsourced to the CSP. The MVT consists of three columns: serial number (*SN*), blocks number (*BN*), and block version (*BV*). The *SN* is an indexing to the file blocks. It indicates the *physical* position of a block in a data file. The *BN* is a counter used to make a *logical* numbering/indexing to the file blocks. Thus, the relation between *BN* and *SN* can be viewed as a mapping between the logical number *BN* and the physical position *SN*.

Remark 2*:* It is important to note that the verifier keeps only *one* table for unlimited number of file copies, *i.e.*, the storage requirement on the verifier side does not depend on the number of file copies on cloud servers. For n copies of a file of size $|F|$, the storage requirement on the CSP side is $O(n|F|)$, while the verifier's overhead is $O(m)$ for all file copies (m is the number of file blocks).

- F is a data file to be outsourced, and is composed of a sequence of m blocks, *i.e.*, $F = \{b_1, b_2, \ldots, b_m\}$.
- $\pi_{key}(\cdot)$ is a pseudo-random permutation (PRP): $key \times \{0,1\}^{\log_2(m)} \rightarrow \{0,1\}^{\log_2(m)}$.[1]
- $\psi_{key}(\cdot)$ is a pseudo-random function (PRF): $key \times \{0,1\}^* \rightarrow \mathbb{Z}_p$ (p is a large prime).
- *Bilinear Map/Pairing:* Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups of prime order p. Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A bilinear pairing is a map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the properties [25]:
  1) *Bilinear:* $\hat{e}(u^a, v^b) = \hat{e}(u,v)^{ab}\ \forall\ u \in \mathbb{G}_1, v \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p$
  2) *Non-Degenerate:* $\hat{e}(g_1, g_2) \neq 1$
  3) *Computable:* there exists an efficient algorithm for computing $\hat{e}$
- $\mathcal{H}(\cdot)$ is a map-to-point hash function : $\{0,1\}^* \rightarrow \mathbb{G}_1$.
- E_K is an encryption algorithm with strong *diffusion* property, *e.g.*, AES.

4 SECURITY ANALYSES

4.1 Security Comparison

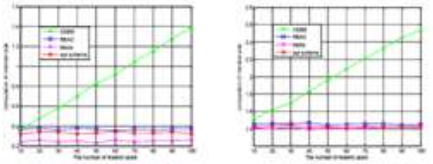
Table 3 Security performance comparison

	Secure key distribution	Access control	Secure user revocation	Anti-collusion attack	Data confidentiality
Mona		√			
RBAC scheme		√			
ODBE		√	√	√	
Our scheme	√	√	√	√	√

In general, our scheme can achieve secure key distribution, fine access control and secure user revocation. For clearly seeing the advantages of security of our proposed scheme, as illustrated in table 3, we list a table compared with Mona, which is Liuet al.'s scheme, the RBAC scheme, which is Zhou et al.'s scheme and ODBE scheme, which is Delerableeet al.'s scheme.The √in the blank meansthe scheme can achieve the corresponding goal.

5 PERFORMANCE EVALUATIONS

We make the performance simulation with NS2 and compare with Mona in [10] and the original dynamic broadcast encryption (ODBE) scheme in [12]. Without loss of generality, we set and the elements in and to be 161 and 1,024 bits, respectively. In addition, we assume the size of the data identity is 16 bits, which yield a group capacity of data files. Similarly, the size of user and group identity are also set 16 bits. Both group members and group managers processes are conducted on a laptop with Core 2 T5800 2.0 GHz, DDR2 800 2G, Ubuntu 12.04 X86. The cloud process is implemented on a laptop with Core i7-3630 2.4 GHz, DDR3 1600 8G, Ubuntu 12.04 X64.



(a) Generating a 10 MB file (b) Generating a 100 MB file

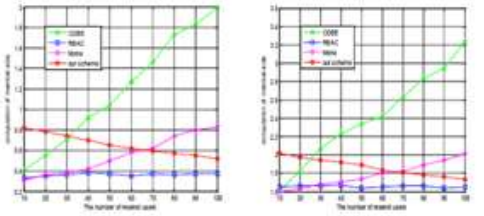
Figure 7 Comparison on computation cost of members for file upload among ODBE, RBAC, Mona and our scheme

As illustrated in figure 7, we list the comparison on computation cost of members for file upload among ODBE, RBAC, Mona and our scheme. It is obviously observed that the computation cost for members in our scheme is irrelevant to the number of revoked users. The reason is that in our scheme, we move the operation of user revocation to the group manager so that the legal clients can encrypt the data files alone without involving information of other clients, including both legal and revoked clients. On the contrary, the computation cost increases with the number of revoked users in ODBE. The reason is that several operations including point multiplications and exponentiations have to be performed by clients to compute the parameters in ODBE.

The computation cost of members for file download operations with the size of 10 and 100Mbytes are illustrated in figure 8. The computation cost is irrelevant to the number of revoked users in RBAC scheme. The reason is that no matter how many users are revoked, the operations for members to decrypt the data files almost remain the same. The computation cost in Mona increases with the number of revoked users,

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

because the users need to perform computing for revocation verification and check whether the data owner is a revoked user. Besides the above operations, more parameters need to be computed by members in ODBE. On the contrary, the computation cost decreases with the number of revoked users in our scheme because of the computation for the recovery of the secret parameter decreases with the number of revoked users.



(a) Downloading a 10 MB file (b) Downloading a 100 MB file

Figure 8 Comparison on computation cost of members for file download among ODBE, RBAC, Mona and our scheme

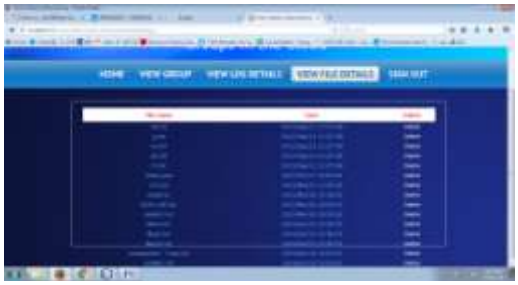
6. RELATED OUT PUTS

Group Manager:

Cloud Groups:

Group Items:

File Details:



File DownLoad:

File Details:

6.2 Future Scope

We design a secure anti-collusion data sharing scheme for dynamic groups in the cloud. In our scheme, the users can securely obtain their private keys from group manager and secure communication channels. Also, our scheme is able

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

to support dynamic groups efficiently, when a new user joins in the group or a user is revoked from the group, the private keys of the other users do not need to be recomputed and updated. Moreover, our scheme can achieve secure user revocation; the revoked users can not be able to get the original data files once they are revoked even if they conspire with the untreated cloud. In this paper I can use identity based encryption algorithm but in future more and new secure encryption technique used and revoked users data may be available but they could not get the original data files.

7. CONCLUSION

In this paper, we design a secure anti-collusion data sharing scheme for dynamic groups in the cloud. In our scheme, the user scan securely obtain their private keys from group manager Certificate Authorities and secure communication channels. Also, our scheme is able to support dynamic groups efficiently, when a new user joins in the group or a user is revoked from the group, the private keys of the other users do not need to be recomputed and updated. Moreover, our scheme can achieve secure user revocation; the revoked users can not be able to get the original data files once they are revoked even if they conspire with the untrusted cloud.

8. REFERENCES

[1] M.Armbrust, A.Fox, R.Griffith, A.D.Joseph, R.Katz, A.Konwinski, G. Lee, D.Patterson, A.Rabkin, I.Stoica, and M.Zaharia. “A View of Cloud Computing,” Comm. ACM, vol. 53, no.4, pp.50-58, Apr.2010.

[2] S.Kamara and K.Lauter, “Cryptographic Cloud Storage,” Proc. Int’l Conf. Financial Cryptography and Data Security (FC), pp.136-149, Jan. 2010.

[3] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K.Fu, “Plutus: Scalable Secure File Sharing on Untrusted Storage,” Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[4] E.Goh, H. Shacham, N. Modadugu, and D. Boneh, “Sirius: Securing Remote Untrusted Storage,” Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.

[6] Shucheng Yu, Cong Wang, Kui Ren, and Weijing Lou, “Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing,” Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

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

Author Name

Lakkimsetti Shanmukha V N Dhana Raju:-B.Tech in C.S.E from affiliated to J.N.T.U. Kakinada. He is pursuing M.Tech in the stream of C.S.E in, Srinivasa Institute of Engineering & Technology an affiliated to J.N.T University Kakinada, A.P, Cheyyeru (v), Amalapuram.

Guide

Saipriya Vissapragada:- Working as Associate professor, Srinivasa Institute of Engineering & Technology affiliated to J.N.T University Kakinada, A.P, Cheyyeru (v), and Amalapuram. Her Qualification is M.tech in C.S.E she has 9 years experience teaching in CSE and she has published 10+ papers on Image Processing, Cloud Computing.