ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Novel Secure Key Generation Protocol for Group Data Sharing

K.Naveed Kumar Reddy
Dept of Computer Science Engineering,
Andhra University College of Engineering,
Visakhapatnam, AP, India.

Dr.Kasukurthi Venkata Rao
Dept of Computer Science Engineering,
Andhra University College of Engineering,
Visakhapatnam, AP, India.

Abstract:
Secure Transmission of data over cloud is still an important research issue in cloud computing and multi owner data sharing takes much importance in recent days of research. We are proposing an efficient data storage mechanism with an efficient group key protocol for secure data transmission between the multi owners and also introducing a novel approach of a new data owner addition without violating the data integrity with authentication of data owners.

Keywords:
Data Sharing, Encryption, Cloud Storage, Key-Aggregate Encryption, Dynamic Group.

1.Introduction:

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 (or a few dollars for more than 1 TB). 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 (e.g., [1]), 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 (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM co-resident with the target one [2]. 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 [3], or without compromising the data owners anonymity [4]. 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, [5], 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.

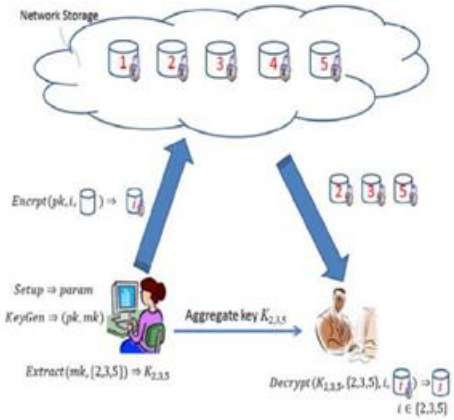

Fig 1. KAC for data sharing in cloud storage.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Naturally, there are two extreme ways for her under the traditional encryption paradigm:

- Alice encrypts all files with a single encryption key and gives Bob the corresponding secret key directly.
- Alice encrypts files with distinct keys and sends Bob the corresponding secret keys.

Obviously, the first method is inadequate since all un-chosen data may be also leaked to Bob. For the second method, there are practical concerns on efficiency. The number of such keys is as many as the number of the shared photos, say, a thousand. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive secure storage. The costs and complexities involved generally increase with the number of the decryption keys to be shared. In short, it is very heavy and costly to do that. 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 has to give the encryptor her secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are different in publickey 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.

2. System Model and Design:

A. System Model

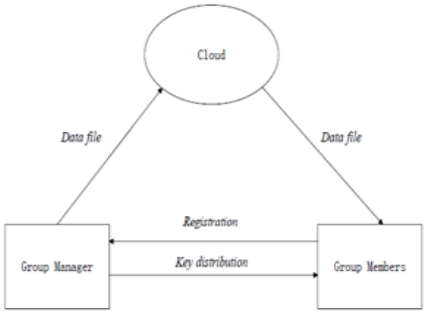

Fig 2. System Model Diagram

As illustrated in figure 2, the system model consists of three different entities: the cloud, a group manager and a large number of group members. The cloud, maintained by the cloud service providers, provides storage space for hosting data files in a pay-as-you-go manner. However, the cloud is untrusted since the cloud service providers are easily to become untrusted. Therefore, the cloud will try to learn the content of the stored data. Group manager takes charge of system parameters generation, user registration,

B. Design:

We describe the main design goals of the proposed scheme including key distribution, data confidentiality, 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.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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. Key Aggregate for Encryption:

We first give the framework and definition for key aggregate encryption. Then we describe how to use KAC
in a scenario of its application in cloud storage.

A. A key-aggregate encryption scheme consists of five polynomial-time algorithms as follows. 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 cipher text 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 delegatees securely (via secure e-mails or secure devices) Finally, any user with an aggregate key can decrypt any cipher text provided that the ciphertext's class is contained in the aggregate key via Decrypt.

a. Setup $(1^{\lambda}, n)$: executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1^{λ} 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 param, which is omitted from the input of the other algorithms for brevity.

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

c. 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.

d. Extract(msk, S): executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegatee. 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 K_S.

e. Decrypt(K_S, 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 $i \in S$.

4. Methodology:

Algorithm:

- Goal is to divide some data D (e.g., the safe combination) into n pieces $D_1, D_2 \ldots D_n$ in such a way that:
  - Knowledge of any k or more D pieces makes D easily computable.
  - Knowledge of anyk -1 or fewer pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).
- This scheme is called (k,n) threshold scheme. If k=n then all participants are required together to reconstruct the secret.
- Suppose we want to use (k,n) threshold scheme to share our secret S where $k < n$.
- Choose at random (k-1) coefficients $a_1, a_2, a_3 \ldots a_{k-1}$, and let S be the a_0
  $f(x)=a_0 + a_1x + a_2x^2 + \ldots\ldots\ldots + a_{k-1}{}^{k-1}$
- Construct n points (i,f(i)) where i=1,2…..n

Example:

- Let S=1234
- n=6 and k=3 and obtain random integers a_1=166 and a_2=94
  $f(x)=1234+166x+94x^2$
- Secret share points (1,1494),(2,1942)(3,2598)(4,3402)(5,4414)(6, 5614)
- We give each participant a different single point (both x and f(x)).

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Re-construction:

- In order to reconstruct the secret any 3 points will be enough
- Let us consider

$(x_0,y_0)=(2,1924),(x_1,y_1)=(4,3402),(x_2,y_2)=(5,4414)$

Using lagrangeous polynomials

$L_0=x-x_1/x_0-x_1*x-x_2/x_0-x_2=x-4/2-4*x-5/2-5=(1/6)x^2-(3/2)x+10/3$

$L_{1=}x-x_0/x_1-x_0*x-x_2/x_1-x_2=x-2/4-2*x-5/4-5=-(1/2)x^2-(7/2)x-5$

$L_2=x-x_0/x_2-x_0*x-x_1/x_2-x_1=x-2/5-2*x-4/5-4=(1/3)x^2-2x+8/3$

$f(x)=\sum_{j=0}^{2} y_j l_j(x)=1942((1/6)x^2-(3/2)x+10/3)+3402(-(1/2)x^2-(7/2)x-)+4414((1/3)x^2-2x+8/3)$

$f(x)=1234+166x+94x^2$

Recall that the secret is the free coefficient, which means that S=1234.

AES algorithm:

AES is a cryptographic algorithm, with this cryptographic algorithm data owner convert the plain data components to cipher with the help of key which is generated from Shamir secret sharing algorithm and uploads the cipher data components to the server and downloads the data components when ever required. A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

5. Performance Evaluation:

Considering that the algorithms including KASE. Setup, KASE. Adjust and KASE. Test are only run in the cloud server, only the execution times in computer are tested.

1) The execution time of KASE. Setup is linear in the maximum number of documents belonging to one owner, and when the maximum number grows up to 20000, it is reasonable that KASE.Setup algorithm only needs 259 second.

2) The execution time of KASE.Encrypt is linear in the number of keywords, and when the number grows up to 10000, KASE.Encrypt algorithm only needs 206 second in computers, but 10018 second in mobile devices. Therefore, we can draw two conclusions; one is that it is not feasible to upload document with lots of keywords using a mobile phone; the other is that the keyword search with pairing computation can be executed quickly in computers now.

3) The execution time of KASE. Extract is linear in the number of shared documents, and when the number grows up to 10000, KASE. Extract algorithm only needs 132 second in computer, but 2430 second in mobile devices. Because the KASE. Extract always runs along with the KASE. Encrypt, it is not suggested to be executed in the mobile devices.

4) The execution time of KASE. Trapdoor is a constant, i.e., 0.01 second in computer and 0.25 second in mobile devices. In fact, the mathematical operation in KASE. Trapdoor is the once multiplication in G, so that the keyword search can be performed efficiently in both mobile devices and computer. Compared with other schemes, there is a significant improvement in our scheme.

5) The execution time of KASE. Adjust is linear in the number of documents. In fact, it can be improved in the practical application, and the details are shown in section 6.4.

6) The execution time of KASE. Test is linear in the number of keyword cipher texts. In fact, the mathematical operation in KASE. Test is twice as much as the pairing computations. When the number grows up to 20000, it will take 467 second.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The below figures show the results after developing the project.

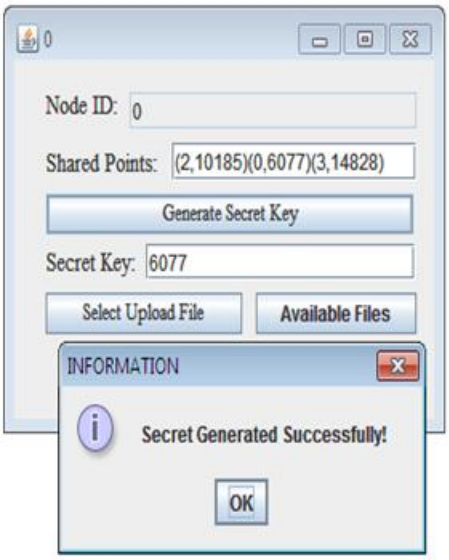

Fig 3. Valid Key generation

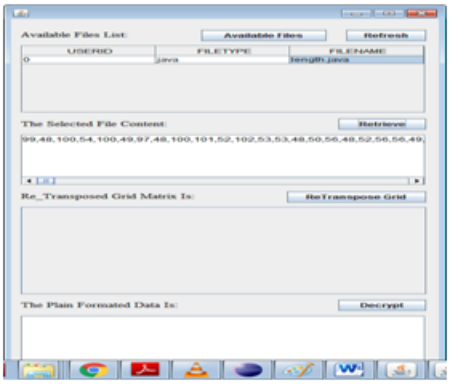

Fig 5: File retrival

6. Conclusion:

This project deals with efficient centralized group key protocol Initially Data member's authentication can be verified with random challenge and secret share. Key can be generated at group key manager and points forwarded to Group members for reconstruction of key and after reconstruction data members verifies the signature or hash code which is applied over key and points for group key manager authentication any group member can be encode and decodes shared files whenever required. We can improve our current research work dynamic member addition and revocation because once session is initiated new user cannot participate in group session but can get the key to encode and decode. Dynamic member addition followed by dynamic key generation improves current research work.

References:

[1] 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.

[2] L. Hardesty, Secure Computers Aren't so Secure. MIT press, http:// www.physorg.com/news 176107396.html, 2009.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[8] S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965- 976, 2012.

[9] D. Boneh, C. G, R. Ostrovsky, G. Persiano. "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522, 2004.

[10] Y. Hwang, P. Lee. "Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System", In: Pairing-Based Cryptography C Pairing 2007, LNCS, pp. 2-22, 2007.

[11] J. Li, Q. Wang, C. Wang. "Fuzzy keyword search over encrypted data in cloud computing", Proc. IEEE INFOCOM, pp. 1-5, 2010.

[12] C. Bosch, R. Brinkma, P. Hartel. "Conjunctive wildcard search over encrypted data", Secure Data Management. LNCS, pp. 114- 127, 2011.

[13] C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers", Journal of Computer Security, pp. 367-397, 2011.

[14] F. Zhao, T. Nishide, K. Sakurai. Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control. Information Security and Cryptology, LNCS, pp. 406-418, 2012.

[15] J. W. Li, J. Li, X. F. Chen, et al. "Efficient Keyword Search over Encrypted Data with Fine-Grained Access Control in Hybrid Cloud", In: Network and System Security 2012, LNCS, pp. 490-502, 2012.

Author's Details:

K.Naveed Kumar Reddy

Pursuing M.Tech in the department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, AP, India. He obtained his B.Tech (CSE) from NBKRIST VIDYA NAGAR.

Dr. Kasukurthi Venkata Rao

M.Tech, Ph.D working as Professor in the department of Computer Science and System Engineering, Andhra University College of Engineering, Visakhapatnam, A.P, India. His research field is in Image Processing, Web Technology, Quantum Cryptography, Data and Cyber Security.