International Journal on Applicationsin Information and Communication Engineering
Volume 2:Issue 6: June 2016, pp 10-13. www.aetsjournal.com ISSN (Online) : 2394 - 6237

Enabling Survey on Aggregate Cryptosystem for Storage Security in Cloud Computing

B.Renugadevi, R.Premkumar

Abstract— Data sharing is an important functionality in cloud storage. In this paper, we show how to securely, efficiently, and flexibly share data with others in cloud storage. We describe new public-key cryptosystems that produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts is possible. 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 cipher text set in cloud storage, but the other encrypted files outside the set remain confidential. The novelty is that one can aggregate any set of secret keys and make them as compact as a single key, but
encompassing the decryption power of all the keys being aggregated. In modern cryptography, a fundamental problem is leveraging the secrecy of a small piece of knowledge into the ability to perform cryptographic functions multiple times. This compact aggregate key can be conveniently sent to others or be stored in asmart card with very limited secure storage. Here the efforts are taken for making the powerful decryption key, which permits decryption of different ciphertexts without expanding the aggregate key size. Here the efforts are taken for making the powerful decryption key, which permits decryption of different ciphertexts without expanding the aggregate key size.

Keywords- Cloud storage, data sharing, ciphertexts, Powerful aggregate key.

I. INTRODUCTION

Automatic face recognition Cloud computing is the new trending model used for computing in which the internet is used for communicating and storing the data. Some of the most crucial functionalities of cloud computing is data sharing and securely storing the important data dumped into cloud.

When it comes to sharing and storing of data, the users of the cloud become bit hesitant to put the data onto the cloud scaring about the confidentiality and security of the data [1]. Due to these aspects of preserving the security and confidentiality of the data, the notion of encryption came into picture. Here the users can encrypt their data using various encryption algorithms before putting them into the cloud. The users can also take the help of the Third Party key generators for encrypting and decrypting of data or can encrypt by themselves using various algorithms[2].

B.Renugadevi, PG Scholar, Department of Computer Science and Engineering, Annai Mathammal Sheela Engineering College, Namakkal, Tamilnadu.(Email: renugabalu90@gmail.com)

R.Premkumar,Assistant Professor,Department of Computer Science and Engineering, Annai Mathammal Sheela Engineering College, Namakkal, Tamilnadu

Cloud storage is day-by-day gaining popularity. It is being utilized as core technology for various online services. In today's world users may apply for free accounts for data sharing, emails, and storing confidential information with storage size up to 25 GB [3]. The wireless technology enables us to access almost all the files, emails and data for the users using their smart devices from any remote corner of the world. Data sharing is a prime functionality in the cloud storage. The blog writers usually allow their friends to have a look or access some of the confidential pictures among the various pictures dumped in the cloud; any organization may grant their employees to access a small part of their confidential data [4]. So here the sharing of the encrypted data with only the authentic users, who are given the rights to access it, is the challenging factor.

Although users have the option of downloading the encrypted data from the cloud, decipher them, and later send them to their friends for sharing it, but this will simply lessen the impact of cloud storage [5]. Instead the authentic users must be given the privilege of rights for accessing while data sharing with others in such a way for accessing those data directly from the server.

Cloud Storage is a service where data is remotely maintained, managed, and backed up. This service is available to users over a network, which is usually the internet. It allows the user to store files online so that the user can access them from any location via the internet [6-8].

The cloud concept that has recently become the technological hot topic is actually very old. It has roots dating back to the 1950's and 1960's. Computer scientist John McCarthy has been credited as one of the founding fathers of the cloud computing concept.

Cloud storage is a subcategory of the very complex cloud computing idea. It is a service model in which data is: maintained, managed and backed up remotely and made available to users over a network (typically the Internet) [9].

Files any where.com was one of the first companies to offer the cloud storage service. Their cloud storage service enables users to store data on their servers from anywhere at any time, while also being able to retrieve the data from anywhere at any time. FilesAnywhere.com would be a pioneer in the cloud storage business and many companies would follow suit [10].

Data sharing functionality is important in cloud storage. Consider Alice has some data to be store in the cloud and does not want expose it to anybody. She first encrypts the data and then uploads in the server in order to avoid data leakage [11-16]. If Bob wants some data of the Alice then he requests her

to share the data. Now the main task is sharing of the encrypted data. There are ways to do this.

1) Alice can encrypt the data using single and share the same key with Bob.

2) Alice can encrypt the data with

II. PROBLEM DEFINATION

The problem of analyzing public sentiment variations and finding the possible reasons causing these variations. To solve the problem, we proposed two Latent Dirichlet Allocation (LDA) based models, Foreground and Background LDA (FB-LDA) and Reason Candidate and Background LDA (RCB-LDA). The FB-LDA model can filter out background topics and then extract foreground topics to reveal possible reasons. To give a more intuitive representation, the RCB-LDA model can rank a set of reason candidates expressed in natural language to provide sentence-level reasons. Another major problem is topic mining. Bulk of opinions consists both foreground and background reasons it is the major challenging issue to differentiate the variations.

A. Scope

To further enhance the readability of the mined reasons, we select the most representative tweets for foreground topics and develop another generative model called Reason Candidate and Background LDA (RCB-LDA) to rank them with respect to their ―popularity within the variation period. Experimental results show that our methods can effectively find foreground topics and rank reason candidates. The proposed models can also be applied to other tasks such as finding topic differences between two sets of documents.

III. METHODOLOGY

The aggregation cryptosystem consists of efficient Key Aggregate Cryptosystem algorithm. The data owner set up the general public parameter using Setup and creates a public/private key and combines using KeyGen. The secret file is encrypted utilizing DES algorithm. The information owner will make use the master-secret to come up with aggregate decipherment key for a collection of data files.

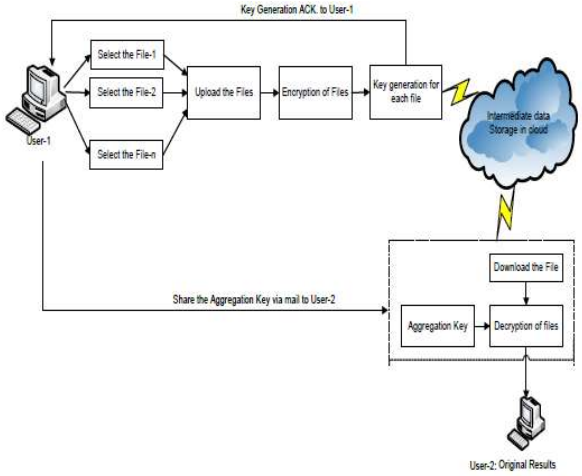

Fig.1 System Architecture

The generated keys may be passed to delegates securely (via secure e-mails or secure devices).

Finally, any user with The Java platform has two Fig. 1. shows the structural behavior of system. In this architecture, the scenario of two users is taken as an example, where the user-1 wants to upload the records onto the cloud, whereas the user-2 wants to download the records from the cloud.

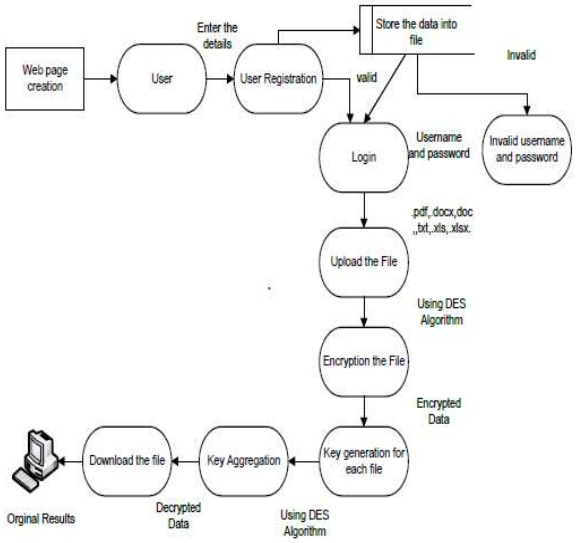

Fig.2 Data flow Diagram

When the user-1 is uploading the records or the files, the data is first encrypted using the DES algorithm and the record gets uploaded onto the cloud. The generated respective private key for each file is displayed as an acknowledgement to the user-1. When the user-2 wants to view or access some files of user-1, he requests the user-1 to share the aggregate key of those particular files, using which after downloading the encrypted files is deciphered using that constant size aggregate key. The user-2 can now download and view all those files using the aggregate key.

An efficient Key-aggregate cryptosystem [1] produce constant size cipher texts such that efficient delegation of decryption rights for any set of cipher text are possible.

IV. NEW USER REGISTRATION

This module contains the fields like Name, Username, Password, Email-id, Mobile number to be entered as a User. In this module all fields must be entered otherwise error message will be displayed. User must register with the cloud then perform the remaining operations. Without registration, other operations cannot be performed. So initially user registers and then goes for the login. In the

user registration form user must enter the valid information otherwise user will get the error message, once the user registers by entering proper information it automatically generates message saying , the user successfully registered after completion.

A. Login

In the user login page, registered user must enter proper user name and password. After that, user performs the further operations. If any error occurs in the user name and password,

an error message is displayed saying invalid username and password.

B. File Upload

Once the user logins successfully, user can upload the file. If the user selected option has an upload file, he then checks the extension of the file. Once user uploads it successfully a message is displayed saying you have successfully uploaded the file. Suppose user uploads a file now, user can encrypt a file by using encryption algorithm and after that the key is generated based on file name, user name, and this encrypted file and key are then stored in the cloud.

C. Key Generation Phase

Once the user uploads the file successfully, user can create key based on file using algorithms. Key will then be stored in the cloud and the filename will also be stored in the cloud system.

D. Requested User

In this module user can request to select the file and key for downloading the required files.

E. Requested Data View

In this module users will view the uploaded files and user can create key based on file using algorithm and the generated aggregate key will be sent through email.

F. Response User

In this module user will get the aggregate key through email and if aggregate key is matched it will verify the filename. Based on file if both values are true, then user can download the file.

G. File Download

If any other person wants to download the file uploaded by the file owner, he requests the key for particular file to the owner. The owner sends the aggregate key of the required files only to the person via email. The person uses this aggregated key to download the file. Then the file is decrypted using decryption algorithm. This decrypted file is then stored in the local system.

The (PBE) Password Based Encryption is a combination of hashing and symmetric encryption, where a 64-bit random number (the salt) is added to the password and hashed using the Message Digest Algorithm i.e. MD5 is input Data files.

Step 1: Setup- In this step the data owner or sender encrypts the data files (Pdf, txt, doc). On input a security level parameter and the number of data file classes n (i.e., each file having private key should be generated by using public and private key pair), it outputs the public system parameter using Data encryption standard algorithms for security purpose.

Step 2: Key Gen- In this step the data owner/ sender randomly generate a public/master-secret key pair (pk, msk).

Step 3: Encrypt- In this step by using above steps the data owner/ sender encrypts the data file by using DES algorithm. The data file is encrypted by using hashing and symmetric key encryption. On input as a data file, a public-key and a private key, it outputs an encrypted data file.

Step 4: KAC- The data owner generates the aggregate key in aggregation cryptosystem by extracting the public private key.

Step 5: Decryption- In this step who wants to download the list of data files using aggregate key, is sent by the data owner/sender to receiver's mail id directly. The receiver after receiving an aggregate key can download the list of data files from the cloud system.

V. Results And Discussions

In the project, the results and discussions are used for comparing the existing system with the proposed model by using the performance analysis plot as shown in Fig. 2. In this below figure, the performance analysis with existing and proposed system by comparing with compression factor and delegation ratio yields the more efficient result than the previous methods.

A. Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed

document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

B. Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs.

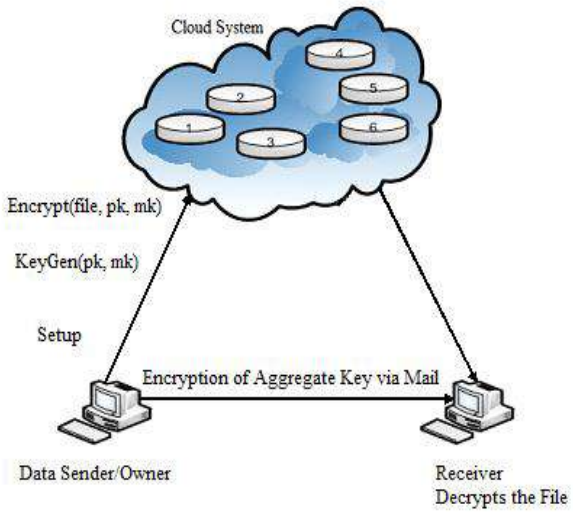

Fig. 3. The Proposed System Framework for Efficient Key Aggregation

In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

VI. Conclusion And Future Work

How to protect users' data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this paper, we consider how to ―compress secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. No matter which one among the power set of classes, the delegate can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

A limitation in our work is the predefined bound of the number of maximum cipher text classes. In cloud storage, the number of cipher texts usually grows rapidly. So we have to reserve enough cipher text classes for the future extension. Otherwise, we need to expand the public-key as we described in Section 4.2. Although the parameter can be downloaded with cipher texts, it would be better if its size is independent of the maximum number of cipher text classes. On the other hand, when one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage, designing a leakage-resilient cryptosystem [22], [34] yet allows efficient and flexible key delegation is also an interesting direction.

References

[1] Cheng-Kang Chu, S. M. Chow, Wen-G Tzeng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", Proc. IEEE Symp. Security and Privacy, Vol. 25, No. 2, Feb. 2014.

[2] K. Kate and S. D. Potdukhe, " Data sharing in cloud storage with key-aggregate cryptosystem", International Journal of Engineering Research and General Science, Volume 2, Issue 6, pp. 882-886, 2014.

[3] S. Prasanna and S. Ramya, "Implementation of Key aggregate Crypto with Stegnography for Secured Data Sharing in Cloud Computing", International Journal of Research in Computer Applications and Robotics, Volume 2, Issue 11, pp. 150-154, 2014.

[4] "Cloud storage", Nonprofit Technology Collaboration, 2013.

[5] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," Cryptology and Network Security , pp. 138-159, 2011.

[6] M. Evans, T. Huynh, K. Le and M. Singh," Cloud Storage", 2011.

[7] S. S. M. Chow, Y. Dodis, Y. Rouselakis and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.

[8] M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009.

[9] B. Alomair and R. Poovendran, "Information Theoretically Secure Encryption with Almost Free Authentication," J. Universal Computer Science, volume 15, Issue 15, pp. 2937-2956, 2009.

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

[11] S.S.M. Chow, J. Weng, Y. Yang and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption," Proc. Progress in Cryptology, Volume 6055, pp. 316-332, 2010.

[12] D. Boneh, C. Gentry and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Advances in Cryptology Conference, Volume 3621, pp. 258-275, 2005.

[13] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology, volume 2139, pp. 213-229, 2001.

[14] G. Ateniese, K. Fu, M. Green, and S. Hohen berger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, Volume 9, Issue 1, pp. 1-30, 2006

[15] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Theory and Applications of Cryptographic Techniques, Volume 3494, pp. 457-473, 2005.

[16] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, Volume 27, Issue 2, pp. 95-98, 1988.