IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

KEY –AGGREGATE SEARCHABLE ENCRYPTION (KASE) FOR GROUP DATA SHARING VIA CLOUD STORAGE

Miss. Priyanka S. Upalanchi*, Prof. S. S.Joshi
* Department of Computer Science & Engineering N.B.Navale Sinhgad College of Engineering,
Solapur 413255

DOI: 10.5281/zenodo.557195

ABSTRACT

Data sharing is an important functionality in cloud storage. In this article, we depict the way of securely, efficiently, and flexibly share data with others in cloud storage. We describe new public-key cryptosystems which produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are 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. This compact aggregate key can be conveniently sent to others or stored in a smart card with very limited secure storage. We provide formal security analysis of our schemes in the standard model. We also describe other application of our schemes. In particular, our schemes give the first public-key patient-controlled encryption for flexible hierarchy, which was yet to be known.

KEYWORDS: public key, cryptosystem, encryption, cloud storage, SHA algorithm, cipher text, drop box.

INTRODUCTION

Cloud storage has emerged as a promising solving a problem for providing ubiquitous, convenient, and on-demand accesses to large amounts of data shared over the Internet. Nowadays, millions of users are sharing personal data, such as photos and videos, with their friends through a dedicated website or other application which enables users to communicate with each and every other by posting information, messages, and images based on cloud storage on a daily basis. Business users are also being attracted by cloud storage due to its too many benefits, including lower cost, greater agility, and better resource utilization. However, while enjoying the quality of being useful, easy, of sharing data via cloud storage, users are also increasingly worried about inadvertent data leaks in the cloud.

Such data leaks, caused by a malicious a misbehaving cloud operator, can usually lead to behave badly break or fail to observe of personal privacy or business secrets (e.g., the recent high probe incident of a famous person photos being leaked in iCloud). To address users relate to protect potential data leaks in cloud storage, a prevalent way of dealing with a situation is for the data owner to encrypt all the data before upload to cloud.

MATERIALS AND METHODS

The best solution for the above problem is that Alice encrypts files with distinct public-keys, but only sends Bob a single (constant-size) decryption key. Since the decryption key should be sent via a secure channel and kept secret, small key size is always desirable. For example, we cannot expect large storage for decryption keys in the resource-constraint devices like smart phones, smart cards or wireless sensor nodes. Especially, these secret keys are usually stored in the tamper-proof memory, which is relatively expensive. The present research efforts mainly focus on minimizing the communication requirements (such as bandwidth, rounds of communication) like aggregate signature. However, not much has been done about the key itself.

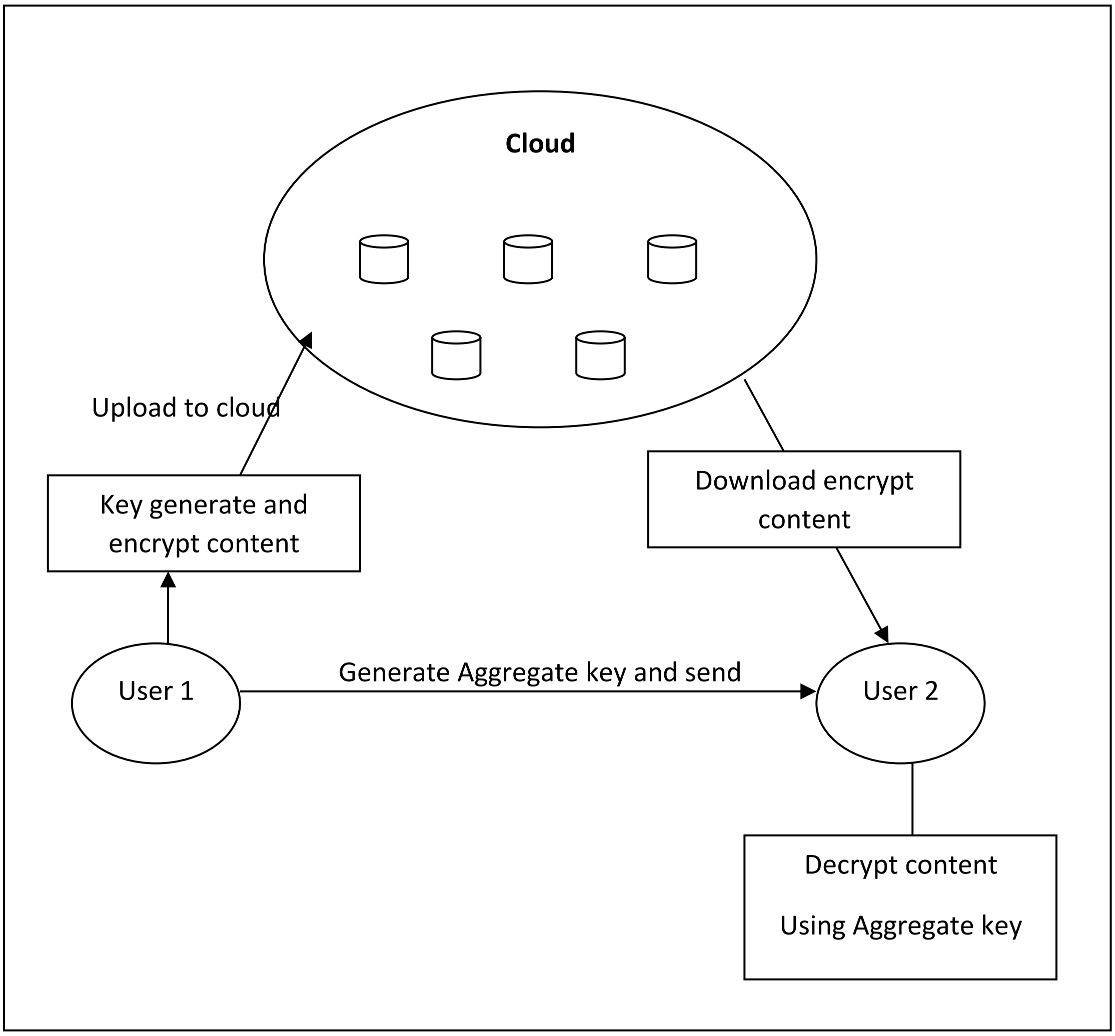

Figure 1: System Architecture

BENEFITS OF PROPOSED SYSTEM

It is more secure.
Decryption key should be sent via a secure channel and kept secret.
It is an efficient public-key encryption scheme which supports flexible delegation.

METHODOLOGY MODULES

Searchable encryption

Generally speaking, searchable encryption schemes fall into two categories, i.e., searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS). Both SSE and PEKS can describe as the tuple SE= (Setup, Encrypt, and Trapdoor Test): Setup (1): this algorithm is run by the owner set up the scheme. It takes as input a security parameter 1, and outputs the necessary keys. Encrypt (k; m): this algorithm is run by the owner to encrypt the data and generate its keyword cipher texts. It takes as input the data m, owner necessary keys including searchable encryption key k and data encryption key, outputs data cipher text and keyword cipher texts C m Trpdr (k; w): this algorithm is run by a user generate a trapdoor Tr for a keyword w using key k. Test (Tr, C): this algorithm is run by the cloud server to perform a keyword search over encrypted data. It takes as input trapdoor Tr and the keyword cipher texts C m, outputs whether C contains the specified keyword.

Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword "urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word "urgent" is a keyword in the email without learning anything else about the email. We refer to this mechanism as Public Key Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. We define the concept of public key encryption with keyword search and give several constructions.

Access control:
Access control is way of limiting access to a system or to physical or virtual resources. In computing, access control is a process by which users are granted access and certain privileges to systems, resources or information .In access control systems, users must present credentials before they can be granted access. In physical systems, these credentials may come in many forms, but credentials that can't be transferred provide the most security. It grants authenticated users access to specific resources based on access policies and the permission level assigned to the user or user group. Access control often includes authentication, which proves the identity of the user or client machine attempting to access the files. After the presentation of the models related to access control in plaintext and encrypted database, we describe how Mute DB transforms an access control matrix for the plaintext model to a matrix suitable for the encrypted database, and how it generates user credentials. Let R be the set of resources that represent plain text tenant data, S the set of plaintext database structures, E the set of encrypted tenant data, U the set of users, and K the set of encryption keys. We define A as the access control matrix where, for each user u P U and for each structure s P S, there exists a binary authorization rule a that defines whether an access to s by u is denied or allowed.

Encrypted database model:
Database encryption is the process of converting data, within a database, In plaintext format into meaningless cipher text by the means of a suitable algorithm. Database decryption is converting the meaningless cipher text into the original information using keys generated by the encryption algorithms. Database encryption be provided at the file or column level. Encryption of a database is costly and requires more storage space than the original data. The steps in encrypting a database are: Determine the criticality of the need for encryption, determine what data needs to be encrypted, determine which algorithms best suit the encryption standard, Determine how the keys will be managed. Numerous algorithms are used for encryption. These algorithms generate keys related to the encrypted data. These keys set a link between the encryption and decryption procedures. The encrypted data can be decrypted only by using these keys. Encrypted data are contained in encrypted tables stored in cloud database servers. For each plaintext table, the Mute DB DBA client generates the corresponding encrypted table and a unique encryption key. The name of the encrypted table is computed by encrypting the name of the plaintext table through that key. The encryption algorithm used for encrypting the table names is a standard AES algorithm in a deterministic mode (e.g., CBC with constant initialization vector).

In such a way, only the users that know the plaintext table name and the corresponding encryption key are able to compute the name of the encrypted table. The deterministic scheme is preferred because it allows a correspondence between plaintext and encrypted tables and improves the efficiency of the query translation process.

Data Group Sharing,
Server can use this aggregate trapdoor and some public information to perform keyword search and return the result to Bob. Therefore, in KASE, the delegation of keyword search right can be achieved by sharing the single aggregate key. We note that the delegation of decryption rights can be achieved using the key-aggregate encryption approach recently proposed, but it remains an open problem to delegate the keyword search rights together with the decryption rights, which is the subject topic of this paper.

Cloud Data Privacy
Cloud Data privacy issues are among the key concerns for companies moving to the cloud. In most countries and in most industries, data privacy regulations apply whenever personally identifiable information (PII) is collected and stored. When this information resides in the cloud, it presents a unique challenge because cloud computing resources are distributed, making it difficult to know where data is located and who has access at any given time. In addition to the cloud data privacy laws highlighted below, many enterprises need to also adhere to series

Cloud Storage

Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. Cloud storage services may be accessed through a co-located cloud computer service, a web service application programming interface (API) or by applications that utilize the API, such as cloud desktop storage, a cloud storage gateway or Web-based content management systems

CONCLUSION

Considering the practical problem of privacy preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents, we for the first time propose the concept of key-aggregate searchable encryption (KASE) and construct a concrete KASE scheme. Both analysis and evaluation results confirm that our work can provide an effective solution to building practical data sharing system based on public cloud storage.

In a KASE scheme, the owner only needs to distribute a single key to a user when sharing lots of documents with the user and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner. However, if a user wants to query over documents shared by multiple owners, he must generate multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi-owners setting is a future work. Moreover, federated clouds have attracted a lot of attention nowadays, but our KASE cannot be applied in this case directly. It is also a future work to provide the solution for KASE in the case of federated clouds.

REFERENCES

[1] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing”,Proc. IEEE INFOCOM, pp. 534-542, 2010.

[2] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure Provenance:The Essential of Bread and Butter of Data Forensics in Cloud Computing”, Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010

[3] X. Song, D.Wagner, A. Perrig. “Practical techniques for searches on encrypted data”, IEEE Symposium on Security and Privacy,IEEE Press, pp. 44C55, 2000.

[4] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. “Searchable symmetric encryption: improved definitions and efficient constructions”, In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.

[5] 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.

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

[7] Z. Liu, Z. Wang, X. Cheng, et al. “Multi-user Searchable Encryption with Coarser-Grained Access Control in Hybrid Cloud”, Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255, 2013.