International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 12, December 2014

A LITERATURE SURVEY ON KEY AGGREGATION SYSTEM FOR SECURE SHARING OF CLOUD DATA

Arun Kumar S, S.Dhanasekar
[1]*PG Student, INFO Institute of Engineering, Coimbatore.*
[2]*Assistant Professor, INFO Institute of Engineering, Coimbatore.*

Abstract—In Cloud computing, data storage is an efficient technique. This survey explains secure, efficient, and flexible method to share data with other people in cloud storage system. This survey, describe novel public-key cryptosystems. This system produce constant-size cipher texts such that efficient delegation of decryption rights for any set of cipher texts are possible. This innovation scheme can aggregate any set of secret keys and make them as a compact single key .The power of all the keys being aggregated in a single key. In other words, holder of the secret key can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage .In this scheme other encrypted files outside the cipher text set remain confidential. The compact aggregate key can be suitably sent to others or be stored in a smart card with very limited secure storage.

Key Words—Cloud storage, data sharing, key-aggregate encryption, patient-controlled encryption

I.INTRODUCTION

Cloud Storage

Cloud storage is gain popularity in recent year. In enterprise settings, demand for data outsourcing is increased today. Data outsourcing should be assists in the strategic management of corporate data. This scheme is also used as a core technology behind many online services. These online services used for online application. Currently this scheme was easy to apply for free accounts for mail, photograph album, sharing of file with storage size more than 25GB. Together by using the current wireless technology, cloud users can access almost all of their files, directories and emails by a mobile phone in any corner of the world.

Data Privacy in Cloud Computing Environment

Considering data privacy in cloud computing environment, a traditional way to ensure data privacy is to rely on the server to enforce the access control after authentication, which means any unexpected privilege increase will expose all data. In a shared-lease cloud computing environment, things become even bad. Data from different users can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VirtualMachine could be stolen by instantiating another Virtual Machine co-occupant with the target one.

ISSN: 2278 – 909X

Data Availability In Cloud Storage

Regarding availability and security of files, there are a more number of cryptographic schemes were proposed. This scheme allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking any information regarding the information, or without compromising the data owner's secrecy.

Cryptography schemes for data storage

Similarly, cloud users will not hold the strong belief that the cloud server is doing a good quality job in terms of privacy. A cryptographic solution, with proven security relied on number-theoretic assumptions is more attractive. Whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. Those users are motivated to encrypt their data with their own keys before uploading the data to the server.

Data sharing in cloud

Data sharing is an important functionality in cloud storage. For example, bloggers can allow their friends view a subset of their private pictures; an enterprise may allow his/her employees access to a portion of susceptible data. The challenging problem is how to efficiently share encrypted data. Users can download the encrypted data from the storage and decrypt them, and then send them to other people for sharing, but it may lose the value of cloud storage. Users can able to give the access rights of the sharing data to other people so that they can access these data from the server openly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial.

Key Sharing Methodology

Based on two methods

- 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 to the corresponding secret keys.

Obviously, the first method is inadequate since all unchosen 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.

Types of Encryption keys

Encryption keys also come with two flavors — symmetric key or asymmetric (public) key.

- **Symmetric Key Encryption**

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.

ISSN: 2278 – 909X

- **Asymmetric Key Encryption**

By contrast, the encryption key and decryption key are different in public-key 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. Therefore, the best solution for the above problem is that Alice encrypts files with separate public-keys, but only sends Bob a single constant-size decryption key. The decryption key should be sent via a secure channel and kept secret. The small key size is always desirable.

For example, we cannot anticipate 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.

II.EXISTING SYSTEM

In 2006 V. Goyal, O. Pandey, A. Sahai, and B. Waters, worked on "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data", this paper develops a new cryptosystem for fine-grained sharing of encrypted data. This scheme was called Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt [4].

Advantages:--

- Applicability of KP-ABE scheme is to sharing of audit-log information and broadcast encryption

In 2007 F. Guo, Y. Mu, Z. Chen, and L. Xu,worked on "Multi-Identity Single-Key Decryption without Random Oracles," This Paper produce Multi-Identity Single-Key Decryption (MISKD).It is an Identity-Based Encryption (IBE) system where a private decryption key can map multiple public keys (identities). More exactly, in MISKD, a single private key can be used to decrypt multiple cipher texts encrypted with different public keys associated to the private key [3].

Advantages

- Multi-Identity Single-Key Decryption scheme is more efficient in decryption.

In 2009 J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, worked on "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," .This system build an efficient system that allows patients both to share partial access rights with others, and to perform searches over their records. We formalize the requirements of a Patient Controlled Encryption scheme, and give several instances, based on existing cryptographic primitives and protocols, each achieving a different set of properties [2].

ISSN: 2278 – 909X

Advantages

- The patient can easily grant access to a category
- Similarly, doctors can add subcategories with arbitrary names, without assistance from the patient. This will be particularly useful if we can't predict the names of all possible subcategories,
- If a doctor needs to add a category for a new type of test, or if categories are labeled by visit dates.

In 2009,M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, worked on "Dynamic and Efficient Key Management for Access Hierarchies,". The proposed solution has the following properties: (i) only hash functions are used for a node to derive a descendant's key from its own key; (ii) the space complexity of the public information is the same as that of storing the hierarchy; (iii) the private information at a class consists of a single key associated with that class; (iv) updates (revocations, additions, etc.) are handled locally in the hierarchy; (v) the scheme is provably secure against collusion; and (vi) key derivation by a node of its descendant's key is bounded by the number of bit operations linear in the length of the path between the nodes[1].

Advantages

- The dynamic scheme achieve a worst- and average-case number of bit operations for key derivation that exponentially better than the depth of a balanced hierarchy.

PROBLEM STATEMENT

- Constant-size decryption key require pre-defined hierarchical relationship.
- The fixed hierarchy is used. In that there is only one way in which we can partition the record. If we want to give out access rights based on something else (e.g. based on document type or sensitivity of data) we will have to look at all the low-level categories involved, and give a separate decryption key for each [2].
- More number of decryption key was used[1].

III. KEY AGGREGATE CRYPTO-SYSTEM

The proposed system design an efficient public-key encryption scheme which supports flexible allocation. In this scheme any subset of the cipher texts (produced by the encryption scheme) is decrypt by a constant-size decryption key (generated by the proprietor of the master-secret key). We solve this problem by introducing a special type of public-key encryption called key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called **class**. Such that cipher texts are further categorized into different classes. The owner of the key holds a master-secret called Master secret key [5].

ISSN: 2278 – 909X

The master-secret can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, such that the decryption power for any subset of cipher text classes. By this solution, Alice can simply send Bob a single aggregate key via a secure channel like email. Bob can download the encrypted photos from Alice's Drop box space and then use this aggregate key to decrypt these encrypted photographs. The scenario is depicted in Figure 1.

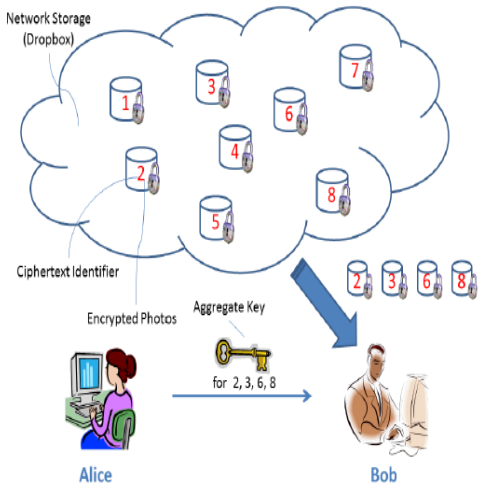

Fig. 1. Alice shares files with identifiers 2, 3, 6 and 8 with Bob by sending him a single aggregate key.

Properties Of KAC

- Decryption key size:- constant.
- Cipher text size:- constant.
- Encryption type:- public-key

Advantages:-

- A decryption key is more powerful in the sense that it allows decryption of multiple cipher texts, without raising its size.
- The size of master-secret key, cipher text, public-key, and aggregate key in our KAC schemes are all are kept constant size.
- KAC scheme is flexible in the sense that there is, no special relation is required between the classes.
- A canonical application of KAC is efficient data sharing scheme.
- The key aggregation property is especially useful when the delegation key to be efficient and flexible.
- The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small cipher text expansion, by distributing to each authorized user a single, compact, small aggregate key.
- The delegation of decryption can be efficiently implemented with the aggregate key.
- Number of cipher text classes is large.
- It is easy to key management.
- Particular Member can view their messages.
- We can provide rigorous security analysis, and extensive performance.

ISSN: 2278 – 909X

Table 1:-Comparative Study on Existing vs. Proposed System

Methods	**Existing System**	**Proposed System**
Technique	• Key-Policy Attribute-Based Encryption (KP-ABE) • Multi-Identity Single-Key Decryption (MISKD)	**Key Aggregate Cryptosystem (KAC)**
Key	**Symmetric**	**Asymmetric Key**
Size Of The Decryption Key	constant-size decryption key	constant-size decryption key
Relationship between Classes	Required	**Not Required**

CONCLUSION

In this survey, we study how to "compress" secret keys in public-key cryptosystems. This compressed key support delegation of secret keys for different cipher text classes in cloud storage. Our approach is more flexible than hierarchical key assignment. The compressed key can only save spaces if all key-holders share a similar set of privileges.

REFERENCES

[1] 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.

[2] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103–114.

[3] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," in Proceedings of Information Security and Cryptology (Inscrypt '07), ser. LNCS, vol. 4990. Springer, 2007, pp. 384–398.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89–98.

[5] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng,, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage," in Proceedings of IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2. Year :2014

ISSN: 2278 – 909X