ISSN (ONLINE): 2349-7084
GLOBAL IMPACT FACTOR 0.238
ISRA JIF 0.351

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, MAY 2015, PP 355-358

A Key Aggregate Construction with Adaptable Offering of Information in Cloud

H.Fareesa Firdose , R.Deepthi Crestose Rebekah

Abstract- Information sharing is an vital utility in cloud storage. In this paper, we show secure, proficient, and adaptable strategy to impart information to other individuals in cloud storage framework. We portray new open key cryptosystems which create fixed size (single key) ciphertexts such that proficient assignments of decryption rights for any set of ciphertexts are conceivable. The novelty is that one can aggregate any arrangement of secret keys and make them as reduced as a solitary (single) key, however including the force of every last one of keys being amassed. As such, the secret key holder can discharge a constant size aggregate key for adaptable decisions of ciphertext set in cloud storage; with our proposed scheme it describes novel adaptive public-key cryptosystems. This reduced total key can be advantageously sent to security channels (e.g email) with exceptionally constrained secure storage space. With our proposed scheme it depicts novel versatile open key cryptosystems. At our previous framework every client can transfer and offer single record with unique index class records with consistent size total key it will turn into a testing issue to lessen the check of cipher class indexes files for mass documents towards enhance the space for more figure class lists .So as to address the above issue we presented a novel versatile key aggregate crypto framework to decrease the no of cipher class indexes for more documents from a same client .This scheme produce constant size cipher text such that proficient assignment of decryption rights for any set of cipher text are possible. This advance scheme can aggregate any set of Secret keys and make them as a minimized single key. The control of all the keys being aggregated in a single key.

Keywords: Aggregate key cryptosystem, Cloud storage, data sharing, key-aggregate encryption.

1. INTRODUCTION:

Cloud storage is nowadays very popular storage system. Cloud storage is storing of data off-site to the physical storage which is maintained by third party. Cloud storage is saving of digital data in logical pool and physical storage spans multiple servers which are manage by third party. Third party is responsible for keeping data available and accessible and physical environment should be protected and running at all time. Instead of storing data to the hard drive or any other local storage, we save data to remote storage which is accessible from anywhere and anytime. It reduces efforts of carrying physical storage to everywhere. By using cloud storage we can access information from any computer through internet which omitted limitation of accessing information from same computer where it is stored. While considering data privacy, we cannot rely on traditional technique of authentication, because unexpected privilege escalation will expose all data. Solution is to encrypt data before uploading to the server with user's own key. Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime to anyone. For example, organization may grant permission to access part of sensitive data to their employees. But challenging task is that how to share encrypted data. Traditional way is user can download the encrypted data from storage, decrypt that data and send it to share with others, but it loses the importance of cloud storage. Cryptography technique can be applied in a two major ways- one is symmetric key encryption and other is asymmetric key encryption. In symmetric key encryption, same keys are used for encryption and decryption. By contrast, in asymmetric key encryption different keys are used, public key for encryption and private key for decryption. Using asymmetric key encryption is more flexible for our approach. This can be illustrated by following example. Suppose Alice put all data on Box.com and she does not want to expose her data to everyone. Due to data leakage possibilities she does not trust on privacy mechanism provided by Box.com, so she encrypt all data before uploading to the server. If Bob ask her to share some data then Alice use share function of Box.com. But problem now is that how to share encrypted data. There are two severe ways: 1. Alice encrypt data with single secret key and share that secret key directly with the Bob. 2. Alice can encrypt data with distinct keys and send Bob corresponding keys to Bob via secure channel. In first approach, unwanted data also get expose to the Bob, which is inadequate. In second approach, no. of keys is as many as no. of shared files, which may be hundred or thousand as well as transferring these keys require secure channel and storage space which can be expensive. Therefore best solution to above problem is Alice encrypts data with distinct public keys, but send single decryption key of constant size to Bob. Since the decryption key should be sent via secure channel and kept secret small size is always enviable. To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertexts

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

• *H.Fareesa Firdose is currently pursuing master's degree program in computer science & engineering in Ravindra College of Engineering for Women(JNTUA), India,**E-mail: fareesafirdose@gmail.com***
• *R.Deepthi Crestose Rebekah,.Currently she is working as Assistant Professor in Department of Computer Science Engineering in Ravindra College of Engineering for Women, Kurnool. India, **E-mail: tonydgs@gmail.com***

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 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.

2. LITERATURE SURVEY

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].
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.

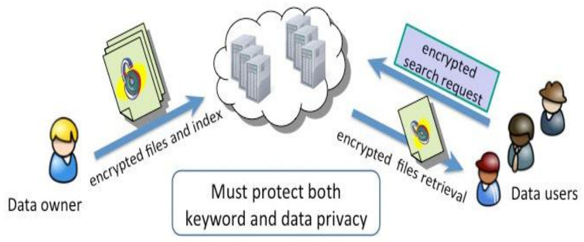
encrypted search request
encrypted files and index
encrypted files retrieval
Data owner
Data users
Must protect both keyword and data privacy

Fig 1 Secured data sharing architecture in cloud

3. OBJECTIVE OF THE SYSTEM

The Objective of the system is to provide best solution for the Existing problem is that Alice encrypts files with distinct public-keys, but only sends Bob a single (constant-size) decryption key while sharing the files cipher text class index also considerable when a same user shares multiple files class index remain constant i.e no variation in class index. Since the decryption key should be sent via a secure channel and kept secret, small key size is always desirable. Using the Public-key Cryptosystem (public key Encryption algorithm)

PROBLEM DEFINITION:-

The challenging problem is how to effectively same user share multiple encrypted files thus class index remains same (constant). Of course users can download the individual or bulk encrypted files from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly

SCOPE:-

We Can 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 for different user and a single cipher text class index remains constant for same user in cloud storage. No matter which one among the power set of classes, the delegatee can always get an aggregate key of constant size.

PROBLEM TESTIMONIAL

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].

4. ASSOCIATED WORK

SYMMETRIC-KEY ENCRYPTION WITH COMPACT KEY AGGREGATE CRYPTOSYSTEM

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 keyaggregate 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].

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.

IBE WITH COMPRESSED KEY

Identity-based encryption (IBE) (e.g., [5], [6], [7]) is a public-key encryption in which the public-key of a user can be set as an identity-string of the user (e.g., an email address, mobile number). There is a private key generator (PKG) in IBE which holds a master-secret key and issues a secret key to each user with respect to the user identity. The content provider can take the public parameter and a user identity to encrypt a message. The recipient can decrypt this ciphertext by his secret key. Guo et al. [8], [9] tried to build IBE with key aggregation. In their schemes, key aggregation is constrained in the sense that all keys to be aggregated must come from different ―identity divisions‖. While there are an exponential number of identities and thus secret keys, only a polynomial number of them can be aggregated.[1] This significantly increases the costs of storing and transmitting ciphertexts, which is impractical in many situations such as shared cloud storage. As Another way to do this is to apply hash function to the string denoting the class, and keep hashing repeatedly until a prime is obtained as the output of the hash function.[1] we mentioned, our schemes feature constant ciphertext size, and their security holds in the standard model. In fuzzy IBE [10], one single compact secret key can decrypt ciphertexts encrypted under many identities which are close in a certain metric space, but not for an arbitrary set of identities and therefore it does not match with our idea of key aggregation.

ATTRIBUTE-BASED ENCRYPTION

Attribute-based encryption (ABE) [11], [12] allows each ciphertext to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a ciphertext can be decrypted by this key if its associated attribute conforms to the policy. For example, with the secret key for the policy $(1 \vee 3 \vee 6 \vee 8)$, one can decrypt ciphertext tagged with class 1, 3, 6 or 8. However, the major concern in ABE is collusion-resistance but not the compactness of secret keys. Indeed, the size of the key often increases linearly with the number of attributes it encompasses, or the ciphertext-size is not constant (e.g., [13]).

PROPOSED STRUCTURE

The data owner establishes the public system parameter through Setup and generates a public/master-secret key pair through KeyGen. Data can be encrypted via Encrypt by anyone who also decides what ciphertext class is associated with the plaintext message to be encrypted. Here data owner can encrypt and share multiple files using same constant cipher text class index towards to reduce the no of Class index files for individual files thus it improve the performance and storage space. The data owner can use the master-secret key pair to generate an aggregate decryption key for a set of ciphertext classes through Extract. The generated keys can be passed to delegates securely through secure e-mails or secure devices Finally, any user with an aggregate key can decrypt any ciphertext provided that the ciphertext's class is contained in the aggregate key via Decrypt. Key aggregate encryption schemes consist of five polynomial time algorithms as follows:

1. SETUP (1Λ , N) : The data owner establish public system parameter via Setup. On input of a security level parameter 1λ and number of ciphertext classes n , it outputs the public system parameter param

2. KEYGEN: It is executed by data owner to randomly generate a public/ master-secret key pair (Pk, msk).

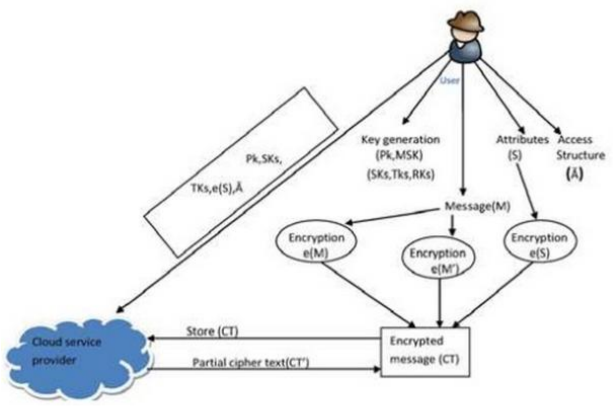

Fig 2. Key Generation Phase

3. ENCRYPT (PK, I, M) : It is executed by data owner and for message m and index i ,it computes the ciphertext as C.

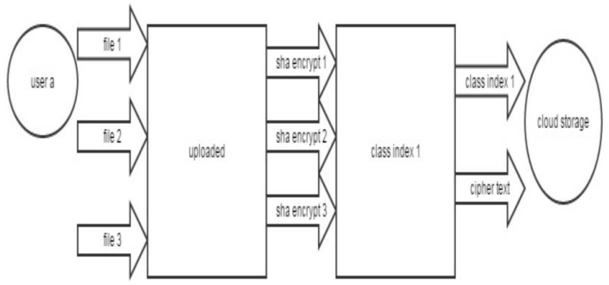

Fig 3. Proposed Architecture

4. EXTRACT (MSK, S): It is executed by data owner for delegating the decrypting power for a certain set of ciphertext classes and it outputs the aggregate key for set S denoted by Ks.

5. DECRYPT (KS, S, I, C): It is executed by a delegate who received, an aggregate key Ks generated by Extract. On input Ks, set S, an index i denoting the ciphertext class ciphertext C belongs to and output is decrypted result m

DATA SHARING

KAC in meant for the data sharing. The data owner can share the data in desired amount with confidentiality. KCA is easy and secure way to transfer the delegation authority.

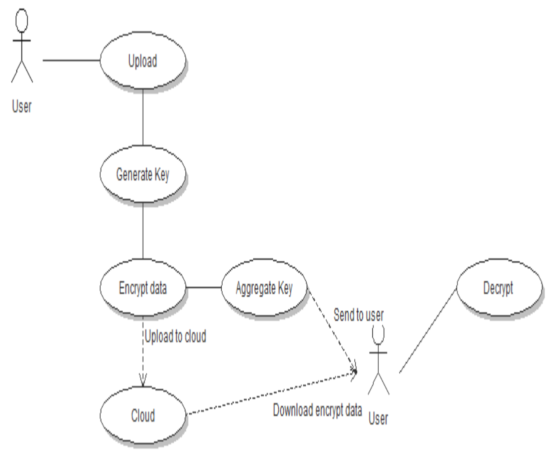

Fig 3. Use Case of Proposed System

For sharing selected data on the server Alice first performs the Setup. Later the public/master key pair (pk, mk) is generated by executing the KeyGen. The msk master key is kept secret and the public key pk and param are made public. Anyone can encrypt the data m and this data is uploaded on server. With the decrypting authority the other users can access those data. If Alice is wants to share a set S of her data with a friend Bob then she can perform the aggregate key KS for Bob by executing Extract (mk, S). As kS is a constant size key and the key can be shared through secure e-mail. When the aggregate key has got Bob can download the data and access it. 5.

PROPERTIES OF KAC

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

5. CONCLUSION

To share information adaptably is basic thing in Cloud computing. Users like to transfer their information on cloud and among distinctive clients. Outsourcing of information to server may prompt release the private information of client to everybody. Encryption is a one deal which gives to impart chose information to fancied competitor. Sharing of decoding keys in secure way assumes vital part. Open key cryptosystems gives appointment of secret keys to distinctive ciphertext classes in distributed storage. The delegate gets safely a aggregate key of consistent size. It is obliged to keep enough number of figure writings classes as they build quick and the ciphertext classes are limited that is the impediment.

REFERENCES

[1] 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, 2013.
[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] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, in Proceedings of Advances in Cryptology - EUROCRYPT '03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.
[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] 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.
[6] M. Chase and S. S. M. Chow, Improving Privacy and Security in Multi-Authority Attribute-Based Encryption, in ACM Conference on Computer and Communications Security, 2009, pp. 121–130
[7] 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.
[8] 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.
[9] 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.
[10] 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.

ABOUT AUTHORS

H.Fareesa Firdose graduated B,Tech in Computer Science and Engineering from G.Pullaiah College of Engineering and Technology(JNTUA) and now pursuing M.Tech (CSE) from Ravindra College of Engineering for Women(JNTUA). Research area of interest includes Cloud Computing, Computer networks.

Mrs .R.DEEPTHI CRESTOSE REBEKAH graduated B.Tech (CSE) from Rajeev Gandhi Memorial College of Engineering & Technology and M.Tech (CSE) from Kottam College of Engineering and Technology Institute of technology & science. Currently she is working as Assistant Professor in Department of Computer Science Engineering in Ravindra college of Engineering for Women, Kurnool. Her areas of interest are cloud computing, wireless sensor networks and computer networks.