IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 12, December 2018, ||Ver. -VI|| PP 12-15

AKC for Scalable Data Sharing In Clouds

[1]Gopalakrishnan, [2]Abinaya suky , [3] Shahanaz S
[1]Assistan Professor, [2], Assistan Professor [3]Assistant Professor
Department of Computer Science and Engineering
CMS College of Engineering and Technology,
Coimbatore, Tamil Nadu, India

Abstract:
Data sharing is an important functionality in cloud storage. To securely, efficiently, and flexibly share data with others in cloud storage. A 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 be stored in a smart card with very limited secure storage. A formal security analysis of our schemes in the standard model is provided.

I. Introduction:

Cloud storage Cloud Storage is a model of networked computer data storage where data is stored on multiple virtual servers, generally hosted by third parties, rather than being hosted on dedicated servers. Hosting companies operate large data centers; and people who require their data to be hosted buy or lease storage capacity from them and use it for their storage needs. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as virtual servers, which the customers can themselves manage. Physically, the resource may span across multiple servers.

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. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, 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. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third2 party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owner's anonymity. 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, with proven security relies 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.

Data sharing is an important functionality in cloud storage. The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data 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. However, finding an efficient and secure way to share partial data in cloud storage is not trivial. 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 encryption her secret key; obviously, this is not always desirable. 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. 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.

Related Works:

A Framework for Secure Data Sharing over Cloud Based on Group Key Management

Now a day's cloud storage gaining more popularity for sharing of data. The sharing of data with more securely, efficiently and flexible through others in the cloud storage. So that by providing security of sharing data we using cryptography technique. In this paper we are using new public key cryptography technique for provide security of data. This paper basically contains two concepts i.e. key generation, encryption and decryption of data. First one is the key generation we are using improved Difficult Hellman key exchange technique. The second one is advanced cryptography technique for data encryption and decryption. So that by proposing those techniques we can provide more secure, efficient and flexible of sharing data.

Analysis and Security based on Attribute based Encryption for data sharing

In Attribute-based Encryption (ABE) scheme, attributes are focused for important role. Attributes are to be segregate to generate a public key for encrypting data and have been used as an access policy to control user's access. The access policy is flavored in two- key policy & cipher text policy. The key policy attribute are used for describing encrypting data and policy implemented in user`s key, and the cipher text policy is the access structure on the cipher text. And the access structure can also be present in either monotonic or non monotonic.

Privacy- Preserving Public Auditing for Secure Cloud Storage

Using cloud storage, users can remotely store their data and enjoy the on demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design

EXISTING SYSTEM:

Besides, in order to improve feasibility and save on the expense in the security paradigm, it is preferred to get the information retrieval result with the most relevant keys that match users interest instead of all the keys, which indicates that the keys should be ranked in the order of relevance by users interest and only the keys with the highest relevance are selected by the users. A series of searchable symmetric encryption schemes have been proposed to enable search on cipher text. Traditional schemes enable users to securely retrieve the cipher text, but these schemes support only Boolean keyword search, i.e., whether a key exists in a system or not, without considering the difference of relevance with the queried keys of these encrypted data in the result. Preventing the security from involving in ranking and entrusting all the work to the user is a natural way to avoid information leakage. However, the limited computational power on the user side and the high computational overhead against information security.

DISADVANTAGES:

To improve security without sacrificing efficiency, schemes presented in show that they support top-k single key retrieval under various scenarios. Authors made attempts to solve the problem of top-k multi-keys over encrypted data. These schemes, however, suffer from two problems. Boolean representation and how to strike a balance between security and efficiency. In the former, data are ranked only by the number of retrieved keys, which impairs search accuracy. In the latter, security is implicitly compromised to tradeoff for efficiency, which is particularly undesirable in security-oriented applications.

PROPOSED SYSTEM:

By utilizing public key based authenticator with random masking privacy preserving public auditing can be achieved. The technique of bilinear aggregate signature is used to achieve key auditing. Key auditing reduces the computation overhead. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient Achieves Key auditing where multiple delegated auditing asks for different keys from different users can be performed simultaneously by the user and also supports dynamic

operations on data blocks i.e. data update, append and delete. We introduce the concepts of similarity relevance and scheme robustness to formulate the privacy issues in encryption schemes, and then solve the insecurity problem by proposing a random key encryption scheme. Novel technologies in the cryptography community and information retrieval community are employed, including encryption and vector space model. In the proposed scheme, the majority of computing work is done on the encrypted data while the user takes part in ranking, which guarantees top k multi-keys provides efficient retrieval of data over encrypted data with high security and practical efficiency.

Advantages:
To achieve better robustness and improve efficiency. This scheme fulfills the secure multi-keyword top-k retrieval over encrypted data. Specifically, for the first time we employ relevance score to support multi-keyword top-k retrieval.

Formally, a FHE scheme consists of five algorithms as follows:
1. Key generation (KG): The algorithm takes as an input a security parameter k and outputs a public and private key pair (pk; sk), where pk is public, while sk is kept secret.
2. Encryption (E): The algorithm takes as input a plaintext m 2 f0; 1g and thepublic key pk, and output a cipher text c, denoted as c ¼ Em;
3. Decryption (D): The algorithm takes as input a cipher text c and the privatekey sk, and outputs a plaintext m 2 f0; 1g, denoted as m ¼ Dc; skÞ.
Information Retrieval:
The generic single database PIR protocol is built on a FHE scheme (KG, E, D, Add, Mult) and consists of three algorithms (Query Generation QG, Response Generation RG, and Response Retrieval RR). At a high level, the user generates a public and private key pair (pk; sk) for the FHE scheme, sends the public key pk to the database server, but keeps the private key sk secret. Then the user chooses an index i, where 1 <i< n, and encrypts i with the public key pk, and sends the cipher text as a query to the database server. Based on the response generation circuit and homomorphic properties, the server computes an encryption of the it bit as a response based on the database, the query and the public key pk, and sends the response back. At the end, the user decrypts the response to obtain the ith bit. Assume that the user and the database server have agreed upon a FHE scheme (KG, E, D, Add, Mult) in advance, our single-database PIR can be using Query generation, Response generation, Response Retrieval.
Hint Text Manipulation:
This system fully involves in the context of generating efficient hint texts against the given data. Once the user inputting the data this system asks the user to provide the hint text for manipulating the data against encryption, after that the hint text and the sampling data will be forwarded to the user's mail for clarification.

Random Key Analysis:
The Random Key Analysis module takes as an input a security parameter k and outputs a public and private key pair (pk; sk), where pk is public, while sk is kept secret. In order to use a smaller set of cryptographic keys, a sender uses multiple keys to encrypt a message and a receiver needs multiple keys to decrypt the message.
Instead of the above mentioned process, this scheme takes a security parameter A and determines a (convenient) parameter set Ap=A,AP=2A,n=(oA2)=r+A, where r is the bit length of the cipher text, n is the bit-length of the secret key, Ap is the bitlength of the noise and oA2 is the number of integers in the public key.
Dynamic Decryption:
In cryptography, encryption is the process of transforming information using an algorithm (called cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information .In many contexts, the word encryption also implicitly refers to the reverse process, decryption to make the encrypted information readable again.

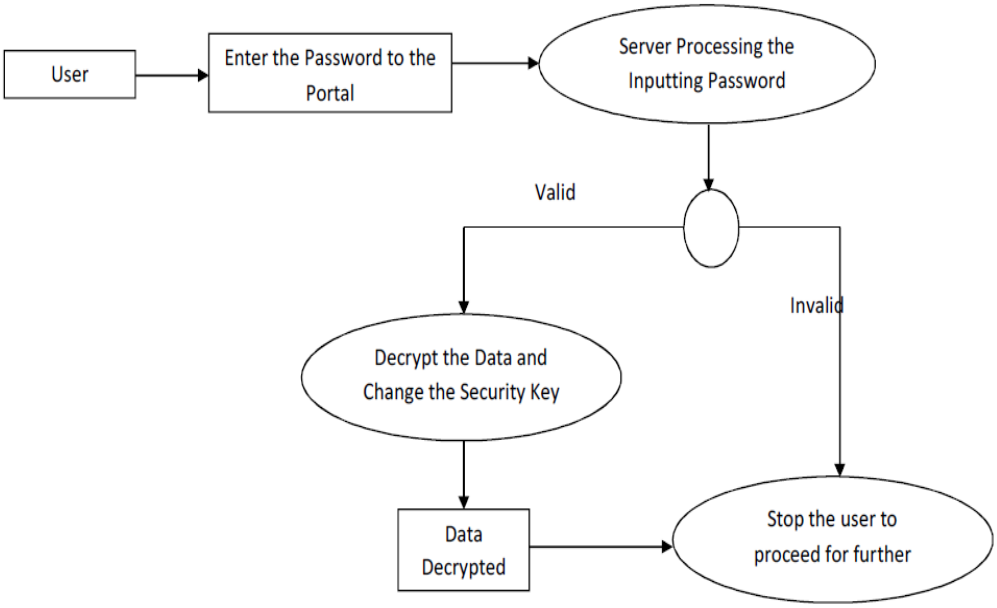

Figure 1: Workflow of the proposed model

II. CONCLUSION:

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 article, 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 delegatee 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. 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 yet allows efficient and flexible key delegation is also an interesting direction.

REFERENCES:

[1]. Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Den "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage" IEEE Transaction feb.2014.

[2]. S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, "SPICE - Simple Privacy- Preserving Identity-Management for Cloud Environment," in Applied Cryptography and Network Security – ACNS 2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.

[3]. L. Hardesty, "Secure computers aren't so secure," MIT press, 2009, http://www.physorg.com/news176107396.html.

[4]. 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.

[5]. B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013. 36

[6]. S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng, "Dynamic Secure Cloud Storage with Provenance," in Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.