KEY AGGREGATE CRYPTOSYSTEM FOR DATA SHARING IN CLOUD STORAGE

Mr. Shubham Raut ,Mr.Mohan Koli ,Mr.Rahul Murkute ,Mr. Ajay Khaire

Mr. Shubham Raut BE(IT),DYPIET(Pimpri),Maharashtra,India

Mr.Mohan Koli BE(IT),DYPIET(Pimpri),Maharashtra,India

Mr.Rahul Murkute BE(IT),DYPIET(Pimpri),Maharashtra,India

Mr. Ajay Khaire BE(IT),DYPIET(Pimpri),Maharashtra,India

ABSTRACT

The important functionality in the cloud storage is data sharing. In this article, we show how to securely, flexibly, and efficiently share data others in cloud storage. Here We describe new technique for 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 new thing is that one can aggregate any set or kind 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 bendable choices of cipher text set in cloud storage, but the other encrypted files outside the set remain secret. This compact aggregate key can be handily sent to others or be stored in a smart card with very limited secure storage. We give the 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.

Key word-*Cloud Storage,Data Sharing ,Key-aggregate encryption,patient-controled encryption*

INTRODUCTION

Cloud storage is gaining more 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 used as a main technology behind many online services for personal applications and services. Nowadays, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world .Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication (e.g., [1]), which means any unexpected privilege escalation will expose all data. In a shared- environment of cloud computing, things become even more 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 steal by another VM co-resident with the target one [2]. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party person to verify the availability of files on behalf of the data owner without leaking anything about the data [3], or without compromising the dataowners secrecy [4]. 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 way out, e.g., [5], with prove security relied on number theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the openness 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. For example, bloggers can allow their friends view a few part of their private pictures; an enterprise may grant her employees access to a portion of sensitive data. The challenging problem is how to successfully share encrypted data. definitely 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 hand over 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 *incomplete* data in cloud storage is not trivial. Below we will take Dropbox1 as an example for illustration. suppose that Alice puts all her private photos on Dropbox, and she does not want to expose her photos to everyone. Due to various data escape possibility Alice cannot feel relieve by just relying on the privacy protection mechanisms provided by Dropbox, so she encrypts all the photos using her own keys before uploading. One day, Alice's friend, Bob, asks her to share the photos taken in all these years which Bob present in. Alice can then use the share function of Dropbox, but the problem now is how to delegate the *decryption rights* for these photos to Bob. A possible option Alice can choose is to strongly send Bob the secret keys involved. Naturally, there are two extreme ways for her under the traditional encryption paradigm:

•Alice encrypts all files with a one encryption key and gives Bob the corresponding secret key directly.

• Alice encrypts files with distinct keys and sends Bob the corresponding secret keys.

Obviously, the first method is not enough since all un chosen data may be also leaked to Bob. For the second method, there are practical concerns on efficiency. The number of photowhich is to be share is same as the the number of keys will be produce, say, a thousand. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive protected 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 types- 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 encryptor her secret key; clearly, 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. For example, in project 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 answer for the above difficulty 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 path and kept secret, small key size is always desirable. For example, we can not expect large storage for decryption keys in the resource-constraint devices like smart phones, smart cards or wireless sensor nodes .Especial, these secret keys are usually stored in the tamper-proof memory, which is relatively expensive The present research efforts mainly centre on minimizing the statement requirements (such as bandwidth, rounds of communication) like aggregate signature [6]. However, not much has been done about the key itself.

REVIEW OF EXISTING WORK

This section reviews the main active work found in the scientific literature that applies Video Live Streaming Over Peer to Peer Network.

[1] Cloud storage could be a storage of information on-line in cloud that is nearby from multiple and connected resources. Cloud storage will offer smart accessibility and reliability, sturdy protection, disaster recovery, and lowest price. Cloud storage having vital practicality i.e. securely, with efficiency, flexibly sharing information with others. New public–key cryptography that is named as Key-aggregate crypto-system (KAC) is introduced. Key-aggregate cryptosystem turn out constant size cipher texts particular economical delegation of decoding rights for any set of Cipher text area unit attainable. Any set of secret keys are often mass and make them as single key, that encompasses power of all the keys being mass. This combination key are often sent to the others for decoding of ciphertext set and remaining encrypted files outside the set area unit remains confidential..

[2] Using cloud storage, users can distantly store their data and enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources, without the load 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 difficult 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 veracity. Thus, enabling public auditability 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 firmly introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online load 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 concurrently 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.

[3] Nowadays, lots of organizations outsource data storage to the cloud such that a member (owner) of an organization can easily share data with other members (users). Due to the existence of security concerns in the cloud, both owners and users are suggested to verify the veracity of cloud data with Provable Data Possession (PDP) before further utilization on data. However, earlier methods either unnecessarily reveal the identity of a data owner to the untrusted cloud or any public verifiers, or introduce significant overheads on verification metadata to preserve secrecy. In this paper, we propose a simple and efficient publicly verifiable approach to ensure cloud data integrity without sacrificing the anonymity of data owners nor requiring significant verification metadata. Specifically, we introduce a security mediator (SEM), which is able to generate verification metadata (i.e., signatures) on outsourced data for data owners. Our approach decouples the secrecy protection mechanism from the PDP. Thus, an organization can employ its own anonymous authentication mechanism, and the cloud is oblivious to that while it only deals with typical PDP-metadata, Consequently, there is no additional storage overhead when compared with existing non-anonymous PDP solutions. The distinctive features of our system also include data privacy, such that the SEM does not learn anything about the data to be uploaded to the cloud at all, which is able to minimize the requirement of trust on the SEM. In addition, we can also extend our system to work with the multi-SEM model, which can avoid the potential single point of failure existing in the single-SEM scenario. Security analyses prove our scheme is secure, and experiment results show our scheme is efficient.

[4] The problem of key organization in an access hierarchy has elicited much interest in the literature. The hierarchy is model as a set of partially ordered classes (represented as a directed graph), and a user who obtains access (i.e., a key) to a sure class can also obtain access to all descendant classes of her class through key derivation. Our answer to the above problem has the following property: (i) only hash function are used for a node to obtain a descendant's key from its own key; (ii) the space complexity of the public in sequence is the same as that of store the hierarchy; (iii) the personal information at a class consists of a single key associated with that class; (iv) updates (revo- cations, additions, etc.) are handled locally in the hierarchy; (v) the scheme is provably protected beside 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 pathway among the nodes. Whereas many earlier schemes had some of these properties, ours is the first that satis-fies all of them. Moreover, for trees (and other "recursively decomposable" hierarchies), we are the first to complete a most horrible- and average-case number of bit operations for key derivation that is exponentially better than the depth of a balanced hierarchy (double-exponentially superior if the hier-archy is unstable, i.e., "tall and skinny"); this is achieved with only a constant increase in the space for the hierarchy. We also confirm how with easy modification our plan can hold extension proposed by Crampton of the stan-dard hierarchies to "limited depth" and reverse inheritance. The security of our scheme relies only on the use of pseudo-random functions.

[5] In cloud, data sharing plays an important role. This survey explains how to securely, efficiently and flexibly distribute the data with others in cloud. This survey depicts a key-aggregate or open key cryptosystem, which produces a stable size ciphertext such that capable delegacy of ciphertext is possible. This new scheme can aggregate several set of secret keys into a solitary key as well as by this solitary key, many files can be decrypted and the files that are outside the ciphertext remain confidential. This aggregate key is stored in a smart card and can be shared with others in a safe channel

[6] As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively mutual just at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for ¯ne-grained sharing of encrypted data that we identify Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that manage

which ciphertexts a client is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption. Our construction supports delegation of personal keys which subsume Hierarchical Identity-Based Encryption (HIBE).

[7]A scheme based on cryptography is proposed for access in a system where hierarchy is represented by a partially order set(or poset).Straightforward implementation of the scheme required users highly place in the hierarchy To store a large number of cryptographic keys .A time-versus-storage trade-off is then described for addressing this key management problem.

[8]We show how to create a master for controlling access to a set of services. Each master key is a concise representation for a list of service key, such that only service key in this list can be computed easily from the master key .Our scheme is more flexible that other, permitting hierarchical organization and expansion of the set of services

[9] The Cryptographic key assignment problem is to assign cryptographic keys to set of partially ordered classes so that the Cryptographic key of a higher class can be used to derive the cryptographic key of a lower class. in this paper we propose a time Cryptographic key of class Ci at a time Kci. key derivation is a constrained not only by class relation time ,but also the time period that is the schemes ,each user hold some secret key parameters whose number is independent number of the classes in the hierarchy and the total time periods. we present to novel application of our scheme. one is broadcast data to authorized user in multilevel security way and the other is to construct flexible key backup system.

[10] A time-bound hierarchical key assignment system is a method to assign time-dependent encryption keys to a set of classes in a partially well-ordered hierarchy, in such a way that each class can calculate the keys of all classes lower down in the hierarchy, according to temporal constraints. In this paper we design and analyze time-bound hierarchical key assignment schemes which are probably-secure and efficient. We consider both the unconditionally secure and the computationally secure settings and distinguish between two different goals: security with respect to key in distinguish ability and against key recovery.

[11] In this paper, we here two *non-zero inner-product* encryption (NIPE) schemes that are *adaptively secure* under a model statement, the decisional linear (DLIN) assumption, in the standard model. The NIPE One of the proposed schemes features *constant-size cipher texts* and *constant-size secret-keys* are the other features. Our NIPE schemes imply an identity-based revocation (IBR) scheme with constant-size cipher texts or constant-size secret-keys that is adaptively safe under the DLIN statement. Any previous IBR scheme with constant size cipher texts or even size secret-keys was *not adaptively secure* in the standard model.
This paper also presents two zero inner-product encryption (ZIPE) scheme each of which has constant-size cipher texts or constant-size secret-keys and is adaptively secure under the DLIN statement in the standard model. They imply an identity-based broadcast encryption (IBBE) system with constant-size cipher texts or constant-size secret-keys that is adaptively secure under the DLIN statement. We also extend the proposed ZIPE schemes into two directions, one is a *fully-attribute-hiding* ZIPE scheme with *constant-size secret-keys*, and the other a *hierarchical* ZIPE scheme with *constant-size cipher texts*. The major idea of our approach is to solve the Fisher's discriminate using deformed kernels incorporating the information of both labelled and unlabeled data. To appraise the effectiveness of our method, we have conducted general experiments on three types of multimedia test beds: the FRGC benchmark face dataset, the Yahoo! web image compilation, and the TRECVID video data collection. The experimental results show that our TKFD algorithm is more successful than traditional supervised approaches, especially when there are very few training data.

[12] Recently, a different type of proxy re-encryption, named conditional proxy re-encryption (C-PRE), has been introduced. Compared with traditional proxy re-encryption, C-PRE enables the delegator to realize fine-grained allocation of decryption rights, and thus is more useful in many applications. In this paper, based on a careful observation on the existing definitions and security notions for C-PRE, we renormalized more accurate definition and security notions for C-PRE. We further propose a more efficient C-PRE scheme, and prove its chosen cipher text protection under the decisional bilinear Diffie-Hellman (DBDH) assumption in the random oracle model. In addition, we point out that a recent C-PRE scheme fails to achieve the chosen-cipher text protection.

[13] Proxy re-encryption (PRE) allows a semi-trusted proxy to translate a cipher text originally intended for Alice into one encrypting the same plaintext for Bob. The proxy only needs are-encryption key given by Alice, and cannot learn anything about the plaintext encrypted. This adds flexibility in various applications, such as classified email, digital right management and distributed storage. In this paper, we study one directional PRE, which the re-encryption key only enables delegation in one direction but not the other direction. In PKC 2009, Shao and Cao proposed a one directional PRE assuming the random oracle. However, we show that it is vulnerable to chosen-cipher text attack (CCA). We then propose an efficient one directional PRE scheme (without resorting to pairings). We gain high efficiency and CCA-protection using the "token-controlled encryption" method, under the computational Diffie-Hellman theory, in the random oracle model and a relaxed but reasonable definition.

[14] In 1998, Blaze, Bleumer, and Strauss (BBS) proposed an application called *atomic proxy re-encryption*, in which a semi-trusted proxy converts a cipher text for Alice into a cipher text for Bob *without* seeing(showing) the underlying plaintext. We predict that quick and secure re-encryption will become increasingly popular as a method for managing encrypted file systems. Although efficiently computable, the wide-spread adoption of BBS re-encryption has been delayed by considerable security risks. next recent work of Do this and Ivan, we present new re-encryption schemes that realize a stronger concept of security, and we demonstrate the usefulness of proxy re-encryption as a method of adding access control to a secure file system. Performance measurements of our experimental file system demonstrate that proxy re-encryption can work effectively in training.

CONCLUSION AND FUTURE WORK

We have existing the modelling of many relationships using CRF for celebrity naming in the Web video domain. In view of the incomplete and noisy metadata, CRF softly encodes these relationships although allow null assignments by considering the uncertainty in labelling. Experimental results basically show that these nice properties lead to performance superiority over several existing approaches. The concern of between video relationships also results in further performance boost, mostly attributed to the capability of rectifying the errors due to missing names and persons. The price of improvement, never the less, also comes along with raise in processing time and the number of false positives. Fortunately, the proposals of leveraging social relation and joint labelling by sequential video processing still make CRF scalable in terms of speed and memory efficiency. While the overall performance of the recommended approach is encouraging, the effectiveness is still limited by facial feature similarity, which is used in the unary energy term and pair wise visual relationship. With the recent advancement in facial feature representations such as Deep Face [23] and face track [33], we plan to investigate the effectiveness of incorporating these representations into the recommended CRF framework in the near future

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "PatientControlled Encryption: Ensuring Privacy of Electronic MedicalRecords," in *Proceedings of ACM Workshop on Cloud ComputingSecurity (CCSW '09)*. ACM, 2009, pp. 103–114.

[6] 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.

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

[8] S. G. Akl and P. D. Taylor, "Cryptographic Solution to a Problemof Access Control in a Hierarchy," *ACM Transactions on ComputerSystems (TOCS)*, vol. 1, no. 3, pp. 239–248, 1983.
[9] G. C. Chick and S. E. Tavares, "Flexible Access Control withMaster Keys," in *Proceedings of Advances in Cryptology - CRYPTO'89*, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.

[10] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 14, no. 1, pp. 182–188, 2002.

[11] G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci,"Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," *J. Cryptology*, vol. 25, no. 2, pp. 243–270, 2012.

[12] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption," in *Cryptology and Network Security (CANS '11)*, 2011,pp. 138–159.

[13] C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional Proxy Broadcast Re- Encryption," in *Australasian Conference on Information Security and Privacy (ACISP '09)*, ser. LNCS, vol. 5594. Springer, 2009, pp. 327–342.

[14] S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng, "EfficientUnidirectional Proxy Re-Encryption," in *Progress in Cryptology -AFRICACRYPT 2010*, ser. LNCS, vol. 6055. Springer, 2010, pp.316–332.

Rahul Dilip Murkute, currently pursuing his BE degree in Information technology from Dr. D. Y. Patil Institute of Engineering and Technology,Pimpri,Pune(Maharashtra)

Ajay Namdeo Khaire, received the diploma in computer engineering from Government polytechnic yavatmal in 2014 and currently pursuing his BE degree in Information technology from Dr. D. Y. Patil Institute of Engineering and Technology,Pimpri,Pune(Maharashtra)

Mohan Ramkrushna Koli, currently pursuing his BE degree in Information technology from Dr. D. Y. Patil Institute of Engineering and Technology,Pimpri,Pune(Maharashtra)

Shubham Sanjay Raut, received the diploma in computer engineering from rambhau lingade polytechnic buldhana in 2014 and currently pursuing his BE degree in Information technology from Dr. D. Y. Patil Institute of Engineering and Technology,Pimpri,Pune(Maharashtra)