International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 09
August 2017

Outsourced Data on Cloud Using Key-Aggregate Crypto System

Vinay Kumar Sinha & P.Anjaiah
M.Tech(Cse) Institute Of Aeronautical Engineering, Hyderabad
Assistant Professor Institute of Aeronautical Engineering, Hyderabad

ABSTRACT

Our Major focus for outsourced data on cloud is how to share these data to other users in secure and flexible way because user don't have physical control over the outsourced data. So Key Aggregate Cryptosystem became that enables the user to set own policies and enforce the policies on the data to be distributed.

KAC allow the user encrypts data and upload on the network storage as well as efficiently delegate the decryption rights for any set of cipher texts. It have more powerful feature which allow the flexible choices of data in cloud storage that compact different encrypted keys for different set of data in a single key i.e. Aggregate Key send to other via mobile OTP, email or to be stored in a little secure storage and remains data will be confidential on cloud. The main goal of KAC have to Access Control, Preventing data access and Integrity of data in network storage.

1. INTRODUCTION

Cloud storage is gaining quality recently. In enterprise settings, we tend to see the increase in demand for information outsourcing, that assists within the strategic management of company information. it's conjointly used as a core technology behind several on-line services for private applications. Nowadays, it's straightforward to use for free of charge accounts for email, image album, file sharing and/or remote access, with storage size over twenty five GB (or some bucks for over one TB). along side this wireless technology, users will access the majority of their files and emails by a itinerant in any corner of the planet. Considering information privacy, a standard thanks to guarantee it's to accept the server to enforce the access management when INTRODUCTION Cloud storage is gaining quality recently. In enterprise settings, we tend to see the increase in demand for information outsourcing, that assists within the strategic management of company information. it's conjointly used as a core technology behind several on-line services for private applications. Nowadays, it's straightforward to use for free of charge accounts for email, image album, file sharing and/or remote access, with storage size over twenty five GB (or some bucks for over one TB). along side this wireless technology, users will access the majority of their files and emails by a itinerant in any corner of the planet. Considering information privacy, a standard thanks to guarantee it's to accept the server to enforce the access management when thanks to share partial information in cloud storage isn't trivial. Below we are going to take Dropbox as AN example for illustration.

Assume that Alice puts all her private photos on Dropbox, and she does not want to expose her photos to everyone. Due to various data leakage possibility Alice cannot feel relieved 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 over all these years which Bob appeared 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 securely 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 single 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 inadequate since all un chosen 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.

2. EXISTING SYSTEM

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.

Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-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 owners 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 relied 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

DISADVANTAGES OF EXISTING SYSTEM:

1. The costs and complexities involved generally increase with the number of the decryption keys to be shared.

2. The encryption key and decryption key are different in public key encryption.

3. PROPOSED SYSTEM

In this paper, we study how to make a decryption key more powerful in the sense that it allows decryption of multiple ciphertexts, without increasing its size. Specifically, our problem statement is “To design an efficient public-key encryption scheme which supports flexible delegation in the sense that any subset of the ciphertexts (produced by the encryption scheme) is decry ptable by a constant-size decryption key (generated by the owner of the master-secret key).” We solve this problem by introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which 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, i.e., the decryption power for any subset of ciphertext classes.

ADVANTAGES OF PROPOSED SYSTEM:

1. The extracted key have can be an aggregate key which is as compact as a secret key for a single class.
2. The delegation of decryption can be efficiently implemented with the aggregate key.

4. IMPLEMENTATION MODULES

- Data Owner(Alice)
- Network Storage
- Aggregate Key Transfer
- User(Bob)

MODULES DESCRIPTION:

Data Owner (Alice): In this module we executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1λ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity.

Network Storage (Drop box): With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Dropbox space and then use this aggregate key to decrypt these encrypted photos. In this Network Storage is untrusted third party server or dropbox.

Aggregate Key Transfer: A key-aggregate encryption scheme consists of five polynomial-time algorithms as follows. The data owner establishes the public system parameter via Setup and generates a public/master-secret key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what ciphertext class is asso-ciated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of ciphertext classes via Extract. The generated keys can be passed to delegates securely (via 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.

User (Bob): The generated keys can be passed to delegates securely (via 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.

5. LITERATURE SURVEY

Simple Privacy-Preserving Identity-Management for Cloud Environment:

S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu. Identity security and privacy have been regarded as one of the top seven cloud security threats. There are a few identity management solutions proposed recently trying to tackle these problems. However, none of these can satisfy all desirable properties. In particular, unlinkability ensures that none of the cloud service providers (CSPs), even if they collude, can link the transactions of the same user. On the other hand, delegatable authentication is unique to the cloud platform, in which several CSPs may join together to provide a packaged service, with one of them being the source provider which interacts with the clients and performs authentication while the others will be transparent to the clients. Note that CSPs may have different authentication mechanisms that rely on different attributes. Moreover, each CSP is limited to see only the attributes that it concerns.

This paper presents SPICE – the first digital identity management system that can satisfy these properties in addition to other desirable properties. The novelty of our scheme stems from combining and exploiting two group signatures so that we can randomize the signature to make the same signature look different for multiple uses of it and hide some parts of the messages which are not the concerns of the CSP. Our scheme is quite applicable to cloud systems due to its simplicity and efficiency.

Privacy-Preserving Public Auditing for Secure Cloud Storage: C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou 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 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 securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards 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.

Storing Shared Data on the Cloud via Security-Mediator: B. Wang, S.S.M. Chow, M. Li, and H. Li Nowadays, many organizations outsource data storage to the cloud such that a member of an organization (data owner) 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 integrity of cloud data with Provable Data Possession (PDP) before further utilization of data. However, previous 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 for preserving anonymity. In this paper, we propose a simple, efficient, and publiclyverifiable approach to ensure cloud data integrity without sacrificing the anonymity of data owners nor requiring significant overhead. 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 anonymity protection mechanism from the PDP. Thus, an organization can employ its own anonymous authentication mechanism, and the cloud is oblivious to that since it only deals with typical PDPmetadata, Consequently, the identity of the data owner is not revealed to the cloud, and there is no extra storage overhead unlike existing anonymous PDP solutions. The distinctive features of our scheme also include data privacy, such that the SEM does not learn anything about the data to be uploaded to the cloud at all, and thus the trust on the SEM is minimized. In addition, we extend our scheme to work with the multi-SEM model, which can avoid the potential single point of failure. Security analyses prove that our scheme is secure, and experiment results demonstrate that our scheme is efficient.

Aggregate and Verifiably Encrypted Signatures from Bilinear Maps: D. Boneh, C. Gentry, B. Lynn, and H. Shacham An aggregate signature scheme is a digital signature that supports aggregation: Given n signatures on n distinct messages from n distinct users, it is possible to aggregate all these signatures into a single short signature. This single signature (and the n original messages) will convince the verifier that the n users did indeed sign the n original messages (i.e., user i signed message Mi for i = 1, . . . , n). In this paper we introduce the concept of an aggregate signature, present security models for such signatures, and give several applications for aggregate signatures. We construct an efficient aggregate signature from a recent short signature scheme based on bilinear maps due to Boneh, Lynn, and

Shacham. Aggregate signatures are useful for reducing the size of certificate chains (by aggregating all signatures in the chain) and for reducing message size in secure routing protocols such as SBGP. We also show that aggregate signatures give rise to verifiably encrypted signatures. Such signatures enable the verifier to test that a given cipher text C is the encryption of a signature on a given message M. verifiably encrypted signatures are used in contract-signing protocols. Finally, we show that similar ideas can be used to extend the short signature scheme to give simple ring signatures.

6. 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 paper, we consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different ciphertext 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 ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension.

Although the parameter can be downloaded with ciphertexts, it would be better if its size is independent of the maximum number of ciphertext 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.

7. REFERENCES

[1] S.S.M. Chow, Y.J. He, L.C.K. Hui, and S.-M. Yiu, "SPICE – Simple Privacy-Preserving IdentityManagement for Cloud Environment," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS), vol. 7341, pp. 526-543, 2012.

[2] L. Hardesty, Secure Computers Aren't so Secure. MIT press, http://www.physorg.com/news176107396.html, 2009.

[3] 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, Feb. 2013.

[4] B. Wang, S.S.M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS), 2013.

[5] S.S.M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R.H. Deng, "Dynamic Secure Cloud Storage with Provenance," Cryptography and Security, pp. 442-464, Springer, 2012.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.

[7] M.J. Atallah, M. Blanton, N. Fazio, and K.B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Trans. Information and System Security, vol. 12, no. 3, pp. 18:1-18:43, 2009.

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

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 09
August 2017

[9] F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles," Proc. Information Security and Cryptology (Inscrypt '07), vol. 4990, pp. 384-398, 2007.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

[11] S.G. Akl and P.D. Taylor, "Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Trans. Computer Systems, vol. 1, no. 3, pp. 239-248, 1983.