ASSURE DEDUPLICATION OF ENCRIPTED DATA IN CLOUD COMPUTING USING ATTRIBUTE BASED STORAGE SYSTEM

[1]Syeda Asma, [2] Mrs. Shashi Rekha H
[1]Mtech in CS&E, [2]Assistant Professor, DOS in CS&E
[1] Department of Studied in Computer Science and Engineering,
[1] VTU PG Centre, Mysore, India

Abstract: In cloud computing where data providers deploy his/her encrypted data to the cloud service providers using an attribute based encryption (ABE), and shares data with the users or client with a specific attribute or credentials. But the ABE does not support the secure deduplication, which leads to critical for eliminating the same data in order to save the storage space and network bandwidth. In this paper we proposed an attribute based storage system with a assured deduplicaton in a hybrid cloud, in this the public cloud manages the storage system, where the private cloud is responsible for duplicate detection. Compare with existing system this system has two advantages. Firstly, rather than sharing the decryption keys, it uses specific access policies to confidentially share data with users. Secondly it access data confidentiality by using standard notation of semantic security, whereas existing system only achieved by using weak security notations, In addition, we put forth a methodology to change a ciphertext over one access policy into ciphertexts of the same plaintext but under other access policies without disclose the underlying plaintext.

Index Terms **- ABE, Storage, Encryptions, ciphertext, plaintext.**

I. INTRODUCTION

Cloud computing extremely facilitates data providers who want to deploy their data to the cloud without disclosing their sensitive data to external parties and would like users with certain credentials to be able to access the data. This requires data to be stored in encrypted forms with access control policies such that no one except users with attributes [1],[2],[3] (or credentials) of specific forms can decrypt the encrypted data. An encryption technique that meets this requirement is called attribute-based encryption (ABE)[4]. where a user's private key is associated with an attribute set, a message is encrypted under an access policy (or access structure) over a set of attributes, and a user can decrypt a ciphertext with his/her private key if his/her set of attributes satisfies the access policy associated with this ciphertext. However, the standard ABE system fails to achieve secure deduplication [5], which is a technique to save storage space and network bandwidth by eliminating redundant copies of the encrypted data stored in the cloud. On the other hand, to the best of our knowledge, existing constructions [6] for secure deduplication are not built on attribute-based encryption. Nevertheless, since ABE and secure deduplication have been widely applied in cloud
computing, it would be desirable to design a cloud storage system possessing both properties.

We consider the following framework in the design of an attribute storage system supporting deduplication of data in cloud ,in this the cloud doesn't store a file or data more than once even though it receives the multiple copies of same data which is encrypted using different access policies. A data provider, Ali, intends to upload a file X to the cloud, and share X with users having certain credentials. In order to do so, Ali encrypts X under an access policy P over a set of attributes, and uploads the corresponding ciphertext to the cloud, such that only users whose sets of attributes satisfying the access policy can decrypt the ciphertext. Later, another data provider, Arun, uploads a ciphertext for the same underlying file X but credit to a different access policy P0. Since the file is uploaded in an encrypted form, the cloud is not able to discern that the plaintext corresponding to Arun's ciphertext is the same as that corresponding to Ali's, and will store X twice. Obviously, such duplicated storage wastes storage space and communication bandwidth. To solve this problem we present an attribute storage system which enroll ciphertext-policy attribute-based encryption(CP-ABE) and comforts assure deduplication.

II. RELATED WORK

2.1 Cloud forensics: State-of-the-art and future directions:

According to K. R. Choo, M. Herman Cloud log forensics (CLF) mitigates the investigation process by identifying the malicious behavior of attackers through profound cloud log analysis. However, the accessibility attributes of cloud logs obstruct accomplishment of the goal to investigate cloud logs for various susceptibilities. Accessibility involves the issues of cloud log access, selection of proper cloud log file, cloud log data integrity, and trustworthiness of cloud logs. Therefore, forensic investigators of cloud log files are dependent on cloud service providers (CSPs) to get access of different cloud logs. Accessing

cloud logs from outside the cloud without depending on the CSP is a challenging research area, whereas the increase in cloud attacks has increased the need for CLF to investigate the malicious activities of attackers. This paper reviews the state of the art of CLF and highlights different challenges and issues involved in investigating cloud log data. The logging mode, the importance of CLF, and cloud log-as-a-service are introduced. Moreover, case studies related to CLF are explained to highlight the practical implementation of cloud log investigation for analyzing malicious behaviors. The CLF security requirements, vulnerability points, and challenges are identified to tolerate different cloud log susceptibilities. We identify and introduce challenges and future directions to highlight open research areas of CLF for motivating investigators, academicians, and researchers to investigate them.

2.2 Google drive: Forensic analysis of data remnants :

According to Quick and K. R. Choo Cloud storage is an emerging challenge to digital forensic examiners. The services are increasingly used by consumers, business, and government, and can potentially store large amounts of data. The retrieval of digital evidence from cloud storage services (particularly from offshore providers) can be a challenge in a digital forensic investigation, due to virtualization, lack of knowledge on location of digital evidence, privacy issues, and legal or jurisdictional boundaries. Google Drive is a popular service, providing users a cost-effective, and in some cases free, ability to access, store, collaborate, and disseminate data. Using Google Drive as a case study, artifacts were identified that are likely to remain after the use of cloud storage, in the context of the experiments, on a computer hard drive and Apple iPhone3G, and the potential access point(s) for digital forensics examiners to secure evidence.

2.3 Fuzzy identity-based encryption:

According to A. Sahai and B. Waters.We introduce a new type of Identity-Based Encryption (IBE) scheme that we call Fuzzy Identity-Based Encryption. In Fuzzy IBE we view an identity as set of descriptive attributes. A Fuzzy IBE scheme allows for a private key for an identity, ω, to decrypt a ciphertext encrypted with an identity, ω', if and only if the identities ω and ω' are close to each other as measured by the "set overlap" distance metric. A Fuzzy IBE scheme can be applied to enable encryption using biometric inputs as identities; the error-tolerance property of a Fuzzy IBE scheme is precisely what allows for the use of biometric identities, which inherently will have some noise each time they are sampled. Additionally, we show that Fuzzy-IBE can be used for a type of application that we term "attribute-based encryption". In this paper we present two constructions of Fuzzy IBE schemes. Our constructions can be viewed as an Identity-Based Encryption of a message under several attributes that compose a (fuzzy) identity. Our IBE schemes are both error-tolerant and secure against collusion attacks. Additionally, our basic construction does not use random oracles. We prove the security of our schemes under the Selective-ID security model.

2.4 Avoiding the disk bottleneck in the data domain

segments and may be forced to access an on-disk index for every input segment. This paper describes three techniques employed in the production Data Domain deduplication file system to relieve the disk bottleneck. These techniques include: (1) the Summary Vector, a compact in-memory data structure for identifying new segments; (2) Stream-Informed Segment Layout, a data layout method to improve on-disk locality for sequentially accessed segments; and (3) Locality Preserved Caching, which maintains the locality of the fingerprints of duplicate segments to achieve high cache hit ratios. Together, they can remove 99% of the disk accesses for deduplication of real world workloads. These techniques enable a modern two-socket dual-core system to run at 90% CPU utilization with only one shelf of 15 disks and achieve 100 MB/sec for single-stream throughput and 210 MB/sec for multi-stream throughput.

2.5 Message-locked encryption and secure deduplication:

According to M. Bellare, S. Keelveedhi We formalize a new cryptographic primitive that we call Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure deduplication (space-efficient secure outsourced storage), a goal currently targeted by numerous cloud storage providers. We provide definitions both for privacy and for a form of integrity that we call tag consistency. Based on this foundation, we make both practical and theoretical contributions. On the practical side, we provide ROM security analyses of a natural family of MLE schemes that includes deployed schemes. On the theoretical side the challenge is standard model solutions, and we make connections with deterministic encryption, hash functions secure on correlated inputs and the sample-then-extract paradigm to deliver schemes under different assumptions and for different classes of message sources. Our work shows that MLE is a primitive of both practical and theoretical interest.

III. PROPOSED SYSTEM

In this paper, we present an attribute-based storage system which employs ciphertext-policy attribute-based encryption (CP-ABE) and supports secure deduplication. Our main contributions can be summarized as follows.

- Firstly, by using hybrid cloud architecture, the system is first to achieve the standard notation of semantic security for data privacy in attribute based deduplication.

- Secondly, we modified a cipher text over one access policy into ciphertexts of same plain text but by using some other access policies without disclosing the underlying plaintext by using forth a methodology.
- This technology might be having independent interest in addition to the approach in the proposed storage system.
- Thirdly, to achieve data consistency in the system, we proceeded towards two cryptographic primitives, such as zero knowledge proof of knowledge and a commitment scheme.

3.1 ADVANTAGES OF PROPOSED SYSTEM:

- We bring in our system a hybrid cloud architecture, which consists of a private cloud responsible for tag checking and ciphertext regeneration and a public cloud storing the ciphertexts.
- Our approach of producing such a proof makes use of the randomness reuse technique in the generation of the tag and the ciphertext with an additional zero-knowledge proof of knowledge (PoK) on the shared random coin in the tag and the ciphertext. Therefore, it is impossible for an adversary to perform duplicate faking attacks unless the adversary casually obtains the content of the plaintext hidden in the ciphertext.

3.2 MODULS OF DECRIPTIONS

Data Provider: In this module, the data provider uploads their report in the cloud server. For the security purpose the data provider encrypts the data file and then store in the cloud. The data provider can change the access policy over data files by attribute based access. The Data provider can have capable of update the encrypted data file. The data provider can set the access privilege to the encrypted data file.

User: In this module, the user can only access the report by access policy and then file access request send to the attribute authority. The encrypted key if the user has the privilege to access the file. For the user level, all the privileges are given by the Attribute authority and the users are controlled by the Attribute Authority only.

Attribute Authority (AA): In this module, the attribute authority view details of data provider and user and then activates his/her account and generate attribute access key. The AA issues every user a decryption key associated with his/her set of attributes. Each user can download an item, and decrypt the ciphertext with the attribute-based private key generated by the AA if this user's attribute set satisfies the access structure.

IND-CPA Security: Attribute-based storage system with secure de-duplication Π. The definition of selective
IND-CPA security with respect to the public cloud in Π, where we restrain algorithm A to issuing queries to the key generation oracle on attribute sets satisfying the access structures A0 and A1.

An attribute-based storage system with secure deduplication Π is IND-CPA secure if the advantage function referring to the security game Game

$$\mathbf{Adv}^{\mathrm{IND}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathrm{def}}{=} \Pr[b' = b]$$

is negligible in the security parameter λ for any probabilistic polynomial-time (PPT) adversary algorithm A.

PRV-CDA Security: Based on the definition of PRV-CDA given the definition of PRV-CDA for Π, Where the adversary is given an additional trapdoor key for the challenge ciphertext but is not given access to any attribute-based private keys (as the private cloud is not allowed to collude with users).
An attribute-based storage system with secure deduplication Π is PRV-CDA secure if the advantage function referring to the security game Game

$$\mathbf{Adv}^{\mathrm{PRV\text{-}CDA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathrm{def}}{=} \Pr[b' = b]$$

is negligible in the security parameter λ for any PPT adversary algorithm

3.3 ARCHITECTURE

The architecture of our attribute-based storage system with secure deduplication is shown in Figure in which four entities are involved: data providers, attribute authority (AA), cloud and users. A data provider wants to outsource his/her data to the cloud and share it with users possessing certain credentials. The AA issues every user a decryption key associated with his/her set of attributes. The cloud consists of a public cloud which is in charge of data storage and a private cloud which performs certain computation such as tag checking.

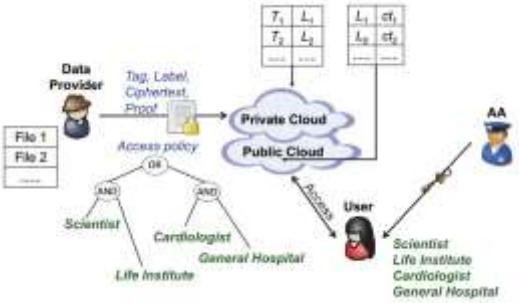

Figure1. System architecture

3.4 **FRAMEWORK**

- Setup(1) ! (pars ,msk). Taking input as security parameters, the setup algorithm outputs the public parameter pars and master private key msk for the system. This algorithem is run by AA.
- KeyGen(pars, msk,A) ! skA. For an attribute set A, the master private key msk and an credential set A as input, this attribute based private key generation algorithm generates an attribute based private key skA, taking the public parameter pars. The algorithm is run by AA.
- Encrypt(pars, M, A) ! (skT , CT). Taking the public parameter pars, we take input as a message M and an access structure A over the universe of attributes , this encryption algorithm outputs a trapdoor key skT and a tuple CT = (T, L, ct, pf), where L and T are the lable and the tag associated with M respectively, ct is the ciphertext which includes the encryption of M as well as the access structure A, and pf is a proof on the relationship of tag T, label L and ciphertext ct. This algorithm is operated by the data provider. Both skT and CT are forwarded to the private cloud. Note that skT cannot be shown to any third party, so it must be sent to the private cloud in a secure manner.
- Validity-Test(pars, CT) ! 1=0. Taking the public parameter pars and a tuple CT as the input, this validity testing algorithm parses CT as (T, L, ct, pf), and outputs 1 if pf is a valid proof for (T, L, ct) or 0 otherwise. This algorithm is run by the private cloud.
- Equality-Test(pars, (T1, L1, ct1), (T2,L2,ct2))! 1=0. Taking the public parameter pars and two tuples (T1, L1, ct1) and (T2, L2, ct2) as the input, this equality testing algorithm outputs 1 if both (T1, L1, ct1), (T2, L2, ct2) are generated from the same underlying message or 0 otherwise. This algorithm is run by the private cloud.
- Re-encrypt(pars, skT , (L, ct), A0) ! (L, ct0). Taking the public parameter pars, the trapdoor key skT , a tag and ciphertext pair (L, ct) and an access structure A0 as the input, this re encryption algorithm outputs a new ciphertext ct0 associated with A0 sharing the same label L of the ciphertext ct0. This algorithm is run by the private cloud.
- Decrypt(pars, (L; ct), **A**, sk**A**) ! M=?. Taking the public parameter pars, a label and ciphertext pair (L; ct) and an attribute-based private key sk**A** associated to an attribute set **A** as the input, this decryption algorithm outputs either the message M when the private key sk**A** satisfies the access structure of the ciphertext ct and the label L is consistent with M (to be defined later), or a symbol ? indicating the failure of the decryption. This algorithm is run by the user.

IV. CONCLUSION

Data providers sends their encrypted data to cloud by users possessing specified attributes attribute based encryption used in cloud computing. The deduplication is an important technique to save storage and network bandwidth, which eliminates identical data, but it does not support secure deduplication. In this paper, we present novel approach of attribute based storage system. Our storage system is built under hybrid cloud architecture; where public cloud manages storage and private cloud manipulate the computation. The private cloud provides the trapdoor key associated with corresponding ciphertext, after receiving request, the private cloud checks the validity of attached proof. If proof matches, the private cloud runs tag matching algorithm. The proposed storage system mainly has two advantages. Firstly, it can be used to privacy share data with other users by particular access policy rather than sharing decryption keys. Secondly, it achieves the standard notation of semantic security while existing deduplication schemes only achieves it by using weaker security notations.

REFERENCES

[1] D. Quick, B. Martini, and K. R. Choo, Cloud Storage Forensics. Syngress Publishing / Elsevier, 2014. [Online]. Available: http://www.elsevier.com/books/cloud-storageforensics/ quick/978-0-12-419970-5

[2] K. R. Choo, J. Domingo-Ferrer, and L. Zhang, “Cloud cryptography: Theory, practice and future research directions,” Future Generation Comp. Syst., vol. 62, pp. 51–53, 2016.

[3] Y. Yang, H. Zhu, H. Lu, J.Weng, Y. Zhang, and K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," Pervasive and Mobile Computing, vol. 28, pp. 122–134, 2016.

[4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, ser. Lecture Notes in Computer Science, vol. 3494. Springer, 2005, pp. 457–473.

[5] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in 6th USENIX Conference on File and Storage Technologies, FAST 2008, February 26- 29, 2008, San Jose, CA, USA. USENIX, 2008, pp. 269–282.

[6] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Advances in Cryptology-EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, ser. Lecture Notes in Computer Science, vol. 7881. Springer, 2013, pp. 296–312.

[7] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, ser. Lecture Notes in Computer Science, vol. 8042. Springer, 2013, pp. 374–391.

[8] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage," in Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013. USENIX Association, 2013, pp. 179–194.

[9] K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud forensics: State-of-the-art and future directions," Digital Investigation, vol. 18, pp. 77–78, 2016.

[10] D. Quick and K. R. Choo, "Google drive: Forensic analysis of data remnants," J. Network and Computer Applications, vol. 40, pp. 179– 193, 2014.