iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: III Month of publication: March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3253

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Storage Supporting Secure Deduplication of Encrypted Data in Cloud

Hareesh ram S[1], Gnana Prakash B[2], Priyadharsini N[3], Veera Lakshmi P[4]
[1,2]Student, [3] Assistant professor, [4] Associate Professor prince shri venkateshwara padmavathy engineering college

***Abstract**: Attribute Based Encryption(ABE) technique is universally used in cloud for storing data. Data providers upload his/her encrypted data in cloud using attribute based encryption technique. But, the standard attribute based encryption technique cannot support secure deduplication. It is very difficult to eliminate duplicate file in cloud. It leads to wastage of bandwidth and memory space in cloud. Due to this, cost of the user is also increased. In this paper, we use hybrid cloud to perform secure deduplication where the private cloud detects the duplicate files and the public cloud is responsible for storing the file. The huffman technique is user to compress the data. Our system has two advantages when compare to prior deduplication methods. Firstly, it achieves the standard notion of semantic security. secondly, without sharing decryption key, it shares confidential data with user specific policies. In add with, we put another method to modify one cipher text over cipher policy into cipher text of same plain text without revealing the plain text.*

***Keywords**: ABE, Deduplication, Storage, Huffman technique, Hybrid cloud.*

I. INTRODUCTION

Cloud computing provides the service to data providers to outsource his/her own data. The data which is uploaded by data providers are in encrypted format. So, without any appropriate credentials the data users will not able to view the data. To access the data, the data user need to provide certain credentials. For this security purpose only the data need to be stored in encrypted format with certain access control policies. So, the user those who are having the access control only decrypts the data. An encryption technique which meets the requirements are called as attribute based encryption. However, the standard ABE technique cannot support secure deduplication. The deduplication is a technique which saves the user memory space and bandwidth. But, the standard ABE technique is widely used in cloud computing. We consider the following scenario in the design of storage supporting secure deduplication of encrypted data in cloud using attribute based technique. The cloud will not store file more than one even though it may receive multiple copies of same file encrypted under different access policies. A data provider, N, intends to upload a file M to the cloud, and share M with users having certain credentials. In order to do so, N encrypts M under an access policy A over a set of attributes, and uploads the corresponding cipher text to the cloud, such that only users whose sets of attributes satisfying the access policy can decrypt the cipher text. Later, another data provider, H, uploads a cipher text for the same underlying file M but ascribed to a different access policy A’. Since the file is uploaded in an encrypted form, the cloud is not able to discern that the plaintext corresponding to H’s cipher text is the same as that corresponding to N’s, and will store M twice. Obviously, such duplicated storage wastes storage space and communication bandwidth. In this paper, we provide attribute based storage supporting secure deduplication(CP-ABE). It supports secure deduplication. Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems. Because it is built based on the hybrid cloud architecture. Secondly, we put forth a methodology to modify a cipher text over one access policy into cipher texts of the same plaintext but under any other access policies without revealing the underlying plaintext. This technique might be of independent interest in addition to the application in the proposed storage system. Thirdly, we propose an approach based on two cryptographic primitives, including a zero-knowledge proof of knowledge and a commitment scheme to achieve data consistency in the system.

The advantages of using this advanced technology are Reducing the Storage Space Faster Recoveries. Effectively, increase network bandwidth, Delete the duplicate files and High Security.

II. RELATED WORK

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.

The paper Security proofs for identity-based identification and signature schemes (2009) provides either security proofs or attacks for a large number of identity-based identification and signature schemes defined either explicitly or implicitly in existing literature. Underlying these are a framework that on the one hand helps explain how these schemes are derived, and on the other hand enables modular security analyses, thereby helping to understand, simplify and unify previous work. The Guillou-Quisquater (GQ) and Schnorr identification schemes(2000) are one among the most efficient and best-known Fiat-Shamir follow-on, but the question of whether they can be proven secure against impersonation under active attack has remained open. This paper provides such a proof for GQ based on the assumed security of RSA under one more inversion, an extension of the usual assumption that was introduced in. It also provides such a proof for the Schnorr scheme based on a corresponding discrete-log related assumption. These are the first security proofs for these schemes under assumptions related to the underlying one-way functions. Both results extend to establish security against impersonation under concurrent attack. The paper Twin clouds An architecture for secure cloud computing. (2011) Cloud computing promises a more cost effective enabling technology to outsource storage and computations. Existing approaches for secure outsourcing of data and arbitrary computations are either based on a single tamper-proof hardware, or based on recently proposed fully homomorphic encryption. The hardware based solutions are not scalable, and fully homomorphic encryption is currently only of theoretical interest and very inefficient. In this paper we propose an architecture for secure outsourcing of data and arbitrary computations to an untrusted commodity cloud. In our approach, the user communicates with a trusted cloud (either a private cloud or built from multiple secure hardware modules) which encrypts and verifies the data stored and operations performed in the untrusted commodity cloud. We split the computations such that the trusted cloud is mostly used for security-critical operations in the less time-critical setup phase, whereas queries to the outsourced data are processed in parallel by the fast commodity cloud on encrypted data. In Reclaiming space from duplicate files in a serverless distributed file system. (2002) The Farsite distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space, it is important to reclaim used space where possible. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes 1) convergent encryption, which enables duplicate files to coalesce into the space of a single file, even if the files are encrypted with different users' keys, and 2) SALAD, a self arranging, lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant.

III. SYSTEM ARCHITECTURE

The architecture Storage Supporting Secure Deduplication of Encrypted Data in Cloud in which four entities are involved: data providers, admin, cloud and users. A data provider wants to outsource his/her data to the cloud and share it with users possessing certain credentials. The admin 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. When sending a file storage request, each data provider firstly creates a tag T and a label L associated with the data, and then encrypts the data under an access structure over a set of attributes. Also, each data provider generates a proof pf on the relationship of the tag T, the label L and the encrypted message ct, but this proof will not best anywhere in the cloud and Is only use during the checking phase for any newly generated storage request. After receiving a storage request, the private cloud first checks the validity of the proof pf, and then tests the equality of the new tag T with existing tags in the system. If there is no match for this new tag T, the private cloud adds the tag T and the label L to a tag-label list, and forwards the label and the encrypted data, (L, ct) to the public cloud for storage. Otherwise, let ct be the cipher text whose tag matches the new tag and L be the label associated with ct, and then the private cloud executes as follows.

A. If the access policy in ct is a subset of that in ct0, the private cloud simply discards the new storage request; else, if the access policy in ct is a subset of that in ct, the private cloud asks the public cloud to replace the stored pair (L0, ct0) with the new pair (L, ct) where L = L0.

B. If the access policies in ct and ct0 are not mutually contained, the private cloud runs the cipher text regeneration algorithm to yield anew cipher text for the same underlying plaintext file and associated with an access structure which is the union of the two access.

The figure 1 which shows the architecture diagram of Storage Supporting Secure Deduplication of Encrypted Data in Cloud. Here, the data provide uploads the file. Before that, he has to sign in the private cloud. The label, file tag, cipher text, proof is loaded in the private cloud. The data user also signs in the private cloud and access the file which is uploaded by the data provider. If the other user uploads the same content in cloud, he has to encrypt the already existed file without revealing the plain text. If data user wants to download the file, he gives the request to the admin. Admin will provides the public key and private key of the corresponding data fie. By using the key, the data user downloads the file. All the processes are done in the private cloud and the public cloud is responsible to the file storage**.**

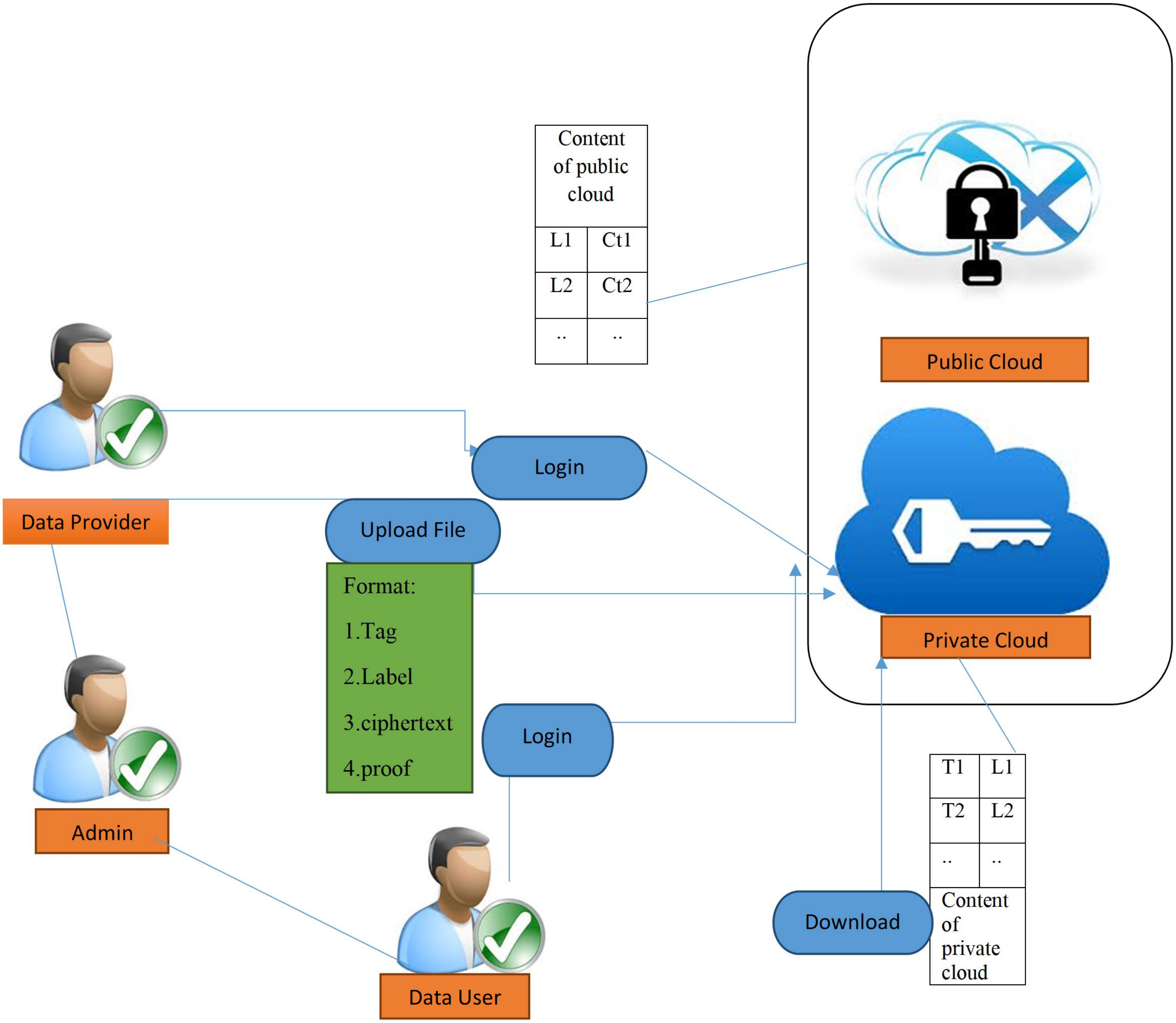

Fig. 1 System architecture for Storage Supporting Secure Deduplication of Encrypted Data in Cloud

IV. MODULE DESCRIPTION

A. Authorization Control Creation and Key Generation:

Authorized user is able to use his/her individual private keys to generate query for certain file and the privileges he/she owned with the help of private cloud, while the public cloud performs duplicate check directly and tells the user if there is any duplicate. duplicate check of any file stored at the S-CSP. In system, the S-CSP is honest but curious and will honestly perform the duplicate check upon receiving the duplicate request from users. The duplicate check token of users should be issued from the private cloud server in our scheme.

It requires that any user without querying the private cloud server for some file token, he cannot get any useful information from the token, which includes the file information or the privilege information.

B. Uploading and Built Hybrid Cloud

In this new deduplication system, a hybrid cloud architecture is introduced to solve the problem. The private keys for privileges will not be issued to users directly, which will be kept and managed by the private cloud server instead. In this way, the users cannot share these private keys of privileges in this proposed construction, which means that it can prevent the privilege key sharing among users in the above straightforward construction. To get a file token, the user needs to send a request to the private cloud server. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user’s identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file. Based on the results of duplicate check, the user either uploads this file or runs PoW.

C. Detect Deduplication

Convergent encryption provides data confidentiality in deduplication. A user derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user also derives a *tag* for the data copy, such that the tag will be used to detect duplicates. Here, we assume that the tag correctness property holds, i.e., if two data copies are the same, then their tags are the same. To detect duplicates, the user first sends the tag to the server side to check if the identical copy has been already stored. Note that both the convergent key and the tag are independently derived and the tag cannot be used to deduce the convergent key and compromise data confidentiality. Both the encrypted data copy and its corresponding tag will be stored on the server side.

D. Key Exchanging

The private keys for the privileges are managed by the private cloud, the file token requests from the users. The interface offered by the private cloud allows user to submit files and queries to be securely stored and computed respectively. The private cloud server will also check the user’s identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file.

E. Verification and File Retrieving

A symmetric key x for each user will be select and set of keys will be sent to the private cloud. An identification protocol equals to proof and verify is also defined, where Proof and Verify are the proof and verification algorithm respectively. In each user *U* is assumed to have a secret key *to* perform the identification with servers. Assume that user *U* has the privilege set *PU*. It also initializes a POW protocol. POW is for the file ownership proof. The private cloud server will maintain a table which stores each user’s public information p kU and its corresponding privilege. It first sends a request and the file name to the S-CSP. Upon receiving the request and file name, the S-CSP will check whether the user is eligible to download file. If failed, the S-CSP sends back an abort signal to the user to indicate the download failure. Otherwise, the S-CSP returns the corresponding cipher text CF. upon receiving the encrypted data from the S-CSP, the user uses the key kF stored locally to recover the original file.

V. CONCLUSION

Attribute-based encryption (ABE) has been widely used in cloud computing where data providers outsource their encrypted data to the cloud and can share the data with users possessing specified credentials. On the other hand, deduplication is an important technique to save the storage space and network bandwidth, which eliminates duplicate copies of identical data. However, the standard ABE systems do not support secure deduplication, which makes them costly to be applied in some commercial storage services. In this paper, we presented a novel approach to realize an attribute-based storage system supporting secure deduplication. Our storage system is built under a hybrid cloud architecture, where a private cloud manipulates the computation and a public cloud manages the storage. The private cloud is provided with a trapdoor key associated with the corresponding cipher text, with which it can transfer the cipher text over one access policy into cipher texts. of the same plaintext under any other access policies without being aware of the underlying plaintext. After receiving a storage request, the private cloud first checks the validity of the uploaded item through the attached proof. If the proof is valid, the private cloud runs a tag matching algorithm to see whether the same data underlying the cipher text has been stored. If so, whenever it is necessary, it regenerates the cipher text into a cipher text of the same plaintext over an access policy which is the union set of both access policies. The proposed storage system enjoys two major advantages. Firstly, it can be used to confidentially share data with other users by specifying an access policy rather than sharing the decryption key. Secondly, it achieves the standard notion of semantic security while existing deduplication schemes only achieve it under a weaker security notion.

REFERENCES

[1] OpenSSL Project. http://www.openssl.org/

[2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX SecuritySymposium, 2013

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013

[5] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009

[6] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002

[7] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshopon Cryptography and Security in Clouds (WCSC 2011), 2011.