iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** V **Month of publication:** May 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Fusion Cloud Method for Protected Authorised Deduplication

Chandragouda.B.H[1], Prof.Anand.S.Uppar[2]
[1,2]*PG student, Head of the department, Department of CSE, SDIT- Mangalore, India*

Abstract--Data de-duplication is one of important data compression techniques for eliminating duplicate copies of same data, and has been used in area of cloud storage to reduce the amount of storage space and saving the more bandwidth. To protect the confidentiality of sensitive data while supporting de-duplication of files, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To protect the security of the data itself, the paper makes the first attempt to address the problem of authorized data de-duplication. Different from other de-duplication systems, the users are further considered in duplicate check besides the data. We also present several new de-duplication constructions supporting authorized duplicate check in fusion cloud architecture. Security analysis explains that our scheme is secure in terms of the definitions specified in the proposed security models. As a proof of concept, we implement a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments using our prototype concept. We show that our proposed authorized de-duplicate check scheme incurs minimal overhead compared to normal operations.
Key words: De-duplication, authorized duplicate check, confidentiality, hybrid cloud

I. INTRODUCTION

Cloud computing provides seemingly unlimited “virtualized” resources to users as services across the whole Internet, while hiding platform and implementation details. Today’s cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified privileges, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data. To make data management scalable in cloud computing, de-duplication has been a well-known technique and has attracted more and more attention recently. Data de-duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, de-duplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. De-duplication can take place at either the file level or the block level. For file level de-duplication, it eliminates duplicate copies of the same file. De-duplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data de-duplication brings a lot of benefits, security and privacy concerns arise as users’ sensitive data are susceptible to both inside and outside attacks. Traditional encryption, while providing data confidentiality, is incompatible with data de-duplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different cipher texts, making de-duplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making de-duplication feasible. It encrypts decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file.

II. LITERATURE SURVEY

P. Anderson and L. Zhang proposed in their paper.Many people now store large quantities of personal and corporate data on laptops or home computers. These often have poor or intermittent connectivity, and are vulnerable to theft or hardware failure. Conventional

backup solutions are not well suited to this environment, and backup regimes are frequently inadequate. This paper describes an algorithm which takes advantage of the data which is common between users to increase the speed of backups, and reduce the storage requirements. This algorithm supports client-end per-user encryption which is necessary for confidential personal data. M. Bellare, S. Keelveedhi, and T. Ristenpart proposed in their paper. We formalize a new cryptographic primitive, 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. M. Bellare, C. Namprempre, and G. Neven proposed in ther paper This paper 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 is 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. We also analyze a generic folklore construction that in particular yields identity-based identification and signature schemes without random oracles.

III. PRELIMINARIES

In this section, we first define the notations used in this paper, review some secure primitives used in our secure de-duplication.

A. *Symmetric Encryption*

Symmetric encryption uses a common secret key κ to encrypt and decrypt information. A symmetric encryption scheme consists of three primitive functions are

KeyGenSE (1_)! κ is the key generation algorithm that generates κ using security parameter 1_;

EncSE (κ, *M*)! *C* is the symmetric encryption algorithm that takes the secret κ and message *M* and then outputs the cipher text *C*; and

DecSE (κ, *C*)! *M* is the symmetric decryption algorithm that takes the secret κ and cipher text *C* and then outputs the original message *M*.

B. *Convergent Encryption*

Convergent encryption provides data confidentiality in de-duplication. A user (or data owner) 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. Both the encrypted data copy and its corresponding tag will be stored on the server side. Formally, a convergent encryption scheme can be defined with four primitive functions:

KeyGenCE (*M*)! *K* is the key generation algorithm that maps a data copy *M* to a convergent key *K*;

EncCE (*K*, *M*) ! *C* is the symmetric encryption algorithm that takes both the convergent key *K* and the data copy *M* as inputs and then outputs a cipher text *C*;

DecCE (*K*, *C*)! *M* is the decryption algorithm that takes both the cipher text *C* and the convergent key *K* as inputs and then outputs the original data copy *M*; and

TagGen(*M*) ! *T* (*M*) is the tag generation algorithm that maps the original data copy *M* and outputs a tag *T*(*M*).

C. *Proof Of Ownership*

The notion of proof of ownership (PoW) enables users to prove their ownership of data copies to the storage server. Specifically, PoW is implemented as an interactive algorithm (denoted by PoW) run by a prover (i.e., user) and a verifier (i.e., storage server). The verifier derives a short value $\phi(M)$ from a data copy *M*. To prove the ownership of the data copy *M*, the prover needs to send ϕ' to the verifier such that $\phi' = \phi(M)$. The formal security definition for PoW roughly follows the threat model in a content distribution network, The accomplices follow the "bounded retrieval model", such that they can help the attacker obtain the file, subject to the constraint that they must send fewer bits than the initial min-entropy of the file to the attacker .

D. Identification Protocol

An identification protocol can be described with two phases: Proof and Verify. In the stage of Proof, a prover/user *U* can demonstrate his identity to a verifier by performing some identification proof related to his identity. The input of the prover/user is his private key *skU* that is sensitive information such as private key of a public key in his certificate or credit card number etc. that he would not like to share with the other users. The verifier performs the verification with input of public information *pkU* related to *skU*. At the conclusion of the protocol, the verifier outputs either accept or reject to denote whether the proof is passed or not. There are many efficient identification protocols in literature, including certificate-based, identity-based identification etc.

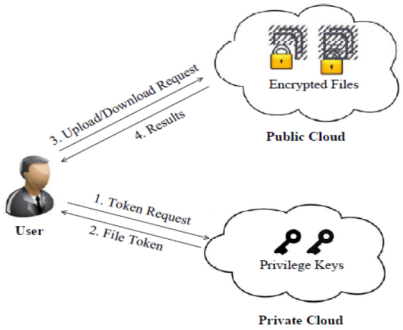


Fig: Architecture for authorized for de-duplication

IV. EXISTING METHODOLOGY

Data de-duplication systems, the private cloud are involved as a proxy to allow data owner/users to securely perform duplicate check with differential privileges.

Such architecture is practical and has attracted much attention from researchers.

The data owners only outsource their data storage by utilizing public cloud while the data operation is managed in private cloud.

V. DISADVANTAGES OF EXISTING SYSTEM

Traditional encryption, while providing data confidentiality, is incompatible with data de-duplication.

Identical data copies of different users will lead to different cipher texts, making de-duplication impossible.

VI. PROPOSED METHODOLOGY

In this paper, we enhance our system in security. Specifically, we present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. In this way, the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the cipher text even collude with the S-CSP. Security analysis demonstrates that our system is secure in terms of the definitions specified in the proposed security model.

VII. ADVANTAGES OF PROPOSED SYSTEM

The user is only allowed to perform the duplicate check for files marked with the corresponding privileges.

We present an advanced scheme to support stronger security by encrypting the file with differential privilege keys.

Reduce the storage size of the tags for integrity check. To enhance the security of de-duplication and protect the data confidentiality,

VIII. APPLICATIONS

Suppress noise.
Canny edge detector is adaptable to various environments.

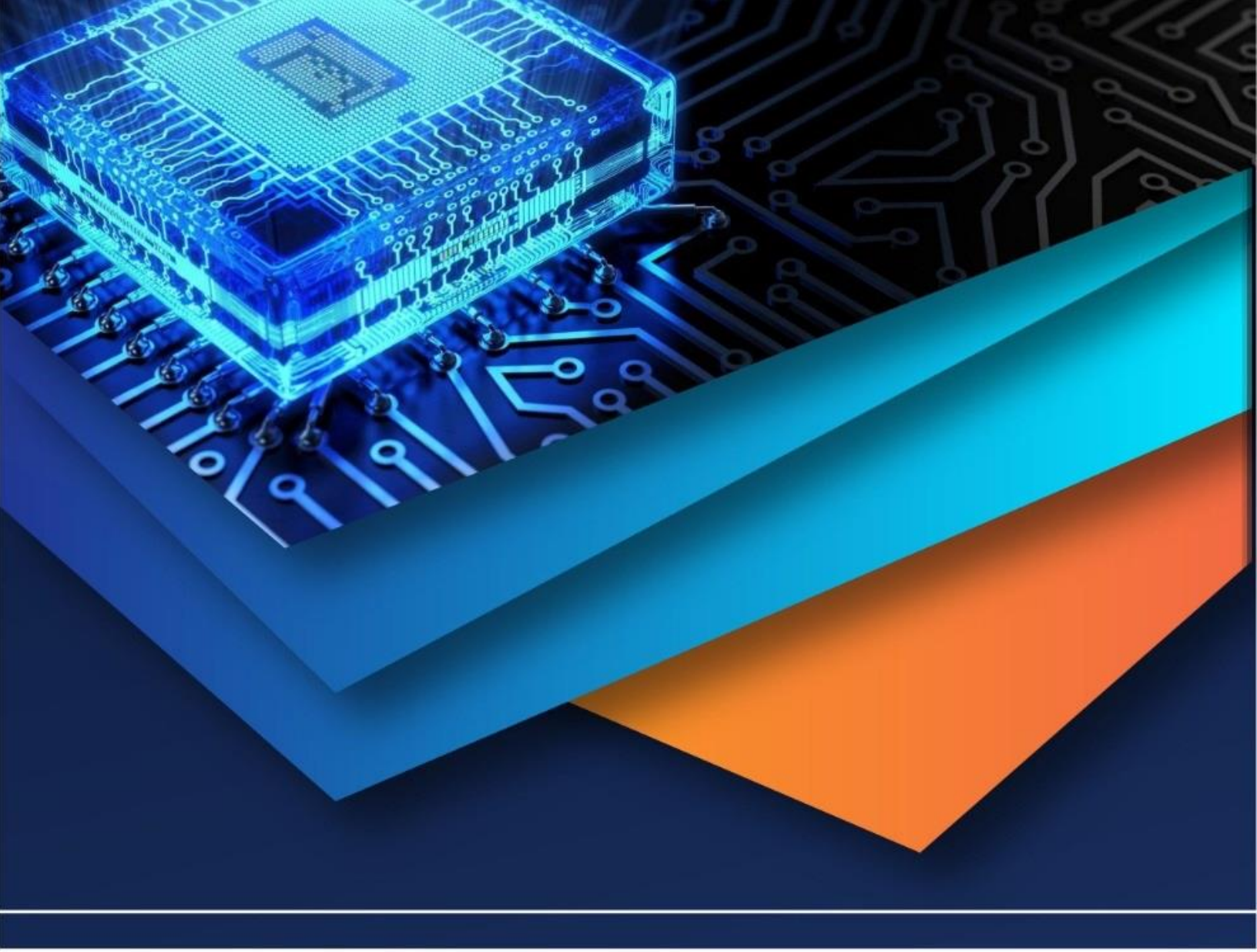
Canny edge detector has been modified in many different ways to solve specific problems.
Robot applications.
The brain MR image analysis in the applications of medicine.

IX. CONCLUSION

In this paper, the notion of authorized data de-duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de-duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

REFERENCES

[1] 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 Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.
[8] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In ICDCS, pages 617–624, 2002.
[9] D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 1992.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY