Secure Deduplication And Data Security With Efficient And Reliable CEKM

N.O.AGRAWAL[1], Prof Mr. S.S.KULKARNI[2]

[1]Department of Information Technology
PRMIT&R, Badnera

[2]Assistant Professor Department of Information Technology
PRMIT&R, Badnera

ABSTRACT

Secure deduplication is a technique for eliminating duplicate copies of storage data, and provides security to them. To reduce storage space and upload bandwidth in cloud storage deduplication has been a well-known technique. For that purpose convergent encryption has been extensively adopt for secure deduplication, critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. The basic idea in this paper is that we can eliminate duplicate copies of storage data and limit the damage of stolen data if we decrease the value of that stolen information to the attacker. This paper makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. To this end, we propose Dekey, User Behaviour Profiling and Decoys technology. Dekey new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers for insider attacker. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments. User profiling and decoys, then, serve two purposes: First one is validating whether data access is authorized when abnormal information access is detected, and second one is that confusing the attacker with bogus information. We posit that the combination of these security features will provide unprecedented levels of security for the deduplication in insider and outsider attacker.

Keywords:- Deduplication, Convergent encryption key management, Dekey, User behaviour profiling, Decoy Technology.

1. INTRODUCTION

Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure deduplication in cloud storage. Although convergent encryption has been extensively adopted for secure deduplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. One critical challenge of today's cloud storage services is the management of the ever-increasing volume of data. To make data management scalable deduplication we are use convergent Encryption for secure deduplication services. Businesses, especially start-ups, small and medium businesses (SMBs), are increasingly opting for outsourcing data and computation to the Cloud. Today's commercial cloud storage services, such as Dropbox, Mozy, and Memopal, have been applying deduplication to user data to save maintenance cost[12] . From a user's point of view, data outsourcing raises security and privacy concerns. We must trust third-party cloud providers to properly enforce confidentiality, integrity checking, and access control mechanisms against any insider and outsider attacks. However, deduplication, while improving storage and bandwidth efficiency, is compatible with Convergent key management. Specifically, traditional encryption requires different users to encrypt their data with their own keys. Many proposals have been made to secure remote data in the Cloud using encryption and standard access controls. It is fair to say all of the standard approaches have been demonstrated to fail from time to time for a variety of reasons, including insider attacks, mis-configured services, faulty implementations, buggy code, and the creative construction of effective and sophisticated attacks not envisioned by the implementers of security procedures. Building a trustworthy cloud computing environment is not enough, because accidents continue to happen, and when they do, and information gets lost, there is no way to get it back. One needs to prepare for such accidents. The basic idea is that we can limit the damage of stolen data if we decrease the value of that stolen information to the attacker. We can achieve this through a 'preventive' disinformation attack. We posit that secure deduplication services can be implemented given two additional security features:

1.1 User Behaviour Profiling: It is expect that access to a user's information in the Cloud will exhibit a normal means of access. User profiling is a well-known technique that can be apply here to model how, when, and how much a user accesses their information in the Cloud. Such 'normal user' behaviour can be continuously check to determine whether abnormal access to a user's information is occurring. This method of behaviour-base security is commonly used in fraud detection applications. Such profiles would naturally include volumetric information, how many documents are typically read and how often. These simple user specific features can serve to detect abnormal Cloud access based partially upon the scale and scope of data transfer [34].

1.2 Decoys: Decoy information, such as decoy documents, honeyfiles, honeypots, and various other bogus information can be generated on demand and serve as a means of detecting unauthorized access to information and to 'poison' the thief 's ex-filtrated information. Serving decoys will confound and confuse an attacker into believing they have bogus useful information, when they have not. Whenever abnormal access to a cloud service is notice, decoy information may be return by the Cloud and deliver in such a way as to appear completely legitimate and normal. The true user, who is the owner of the information, would readily identify when decoy information is being return by the Cloud, and hence could alter the Cloud's responses through a variety of means, such as challenge questions, to inform the Cloud security system that it has inaccurately detect an abnormal access. In the case where the access is correctly identified as an abnormal access, the Cloud security system would deliver unbounded amounts of bogus information to the adversary, thus securing the user's true data from unauthorized disclosure.

The decoys, then, serve two purposes:

First is that validating whether data access is authorized when abnormal information access is detected, and second one is that confusing the attacker with bogus information.

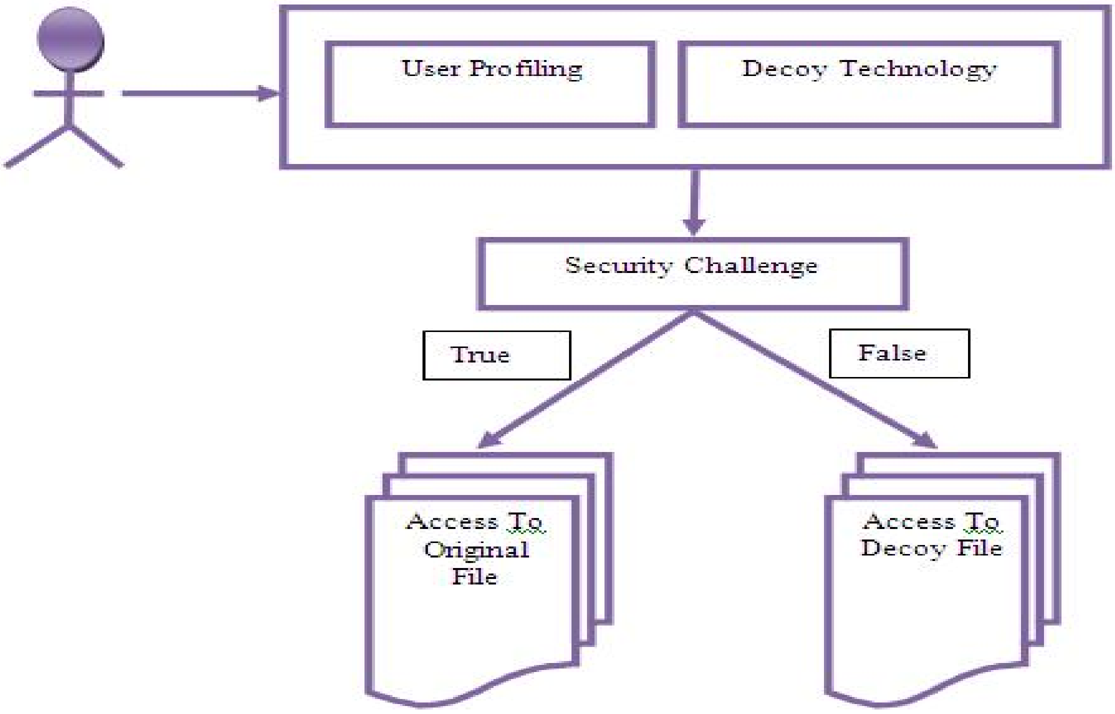

Fig: Outsider Attacker data security

Now this is all about of outsider attacker protection while increase insider attacker with secure deduplication we are use convergent encryption [8] provides a viable option to enforce data confidentiality while realizing deduplication. It encrypts/decrypts a data copy with a convergent key, which is derived by computing the cryptographic hash value of the content of the data copy itself [8]. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since encryption is deterministic, identical data copies will generate the same convergent key and the same cipher text. This allows the cloud to perform deduplication on the cipher texts. The cipher text scan only is decrypted by the corresponding data owners with their convergent keys. Dekey new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers for insider attacker. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments. We posit that the combination of these security features will provide unpredictable levels of security for the deduplication.

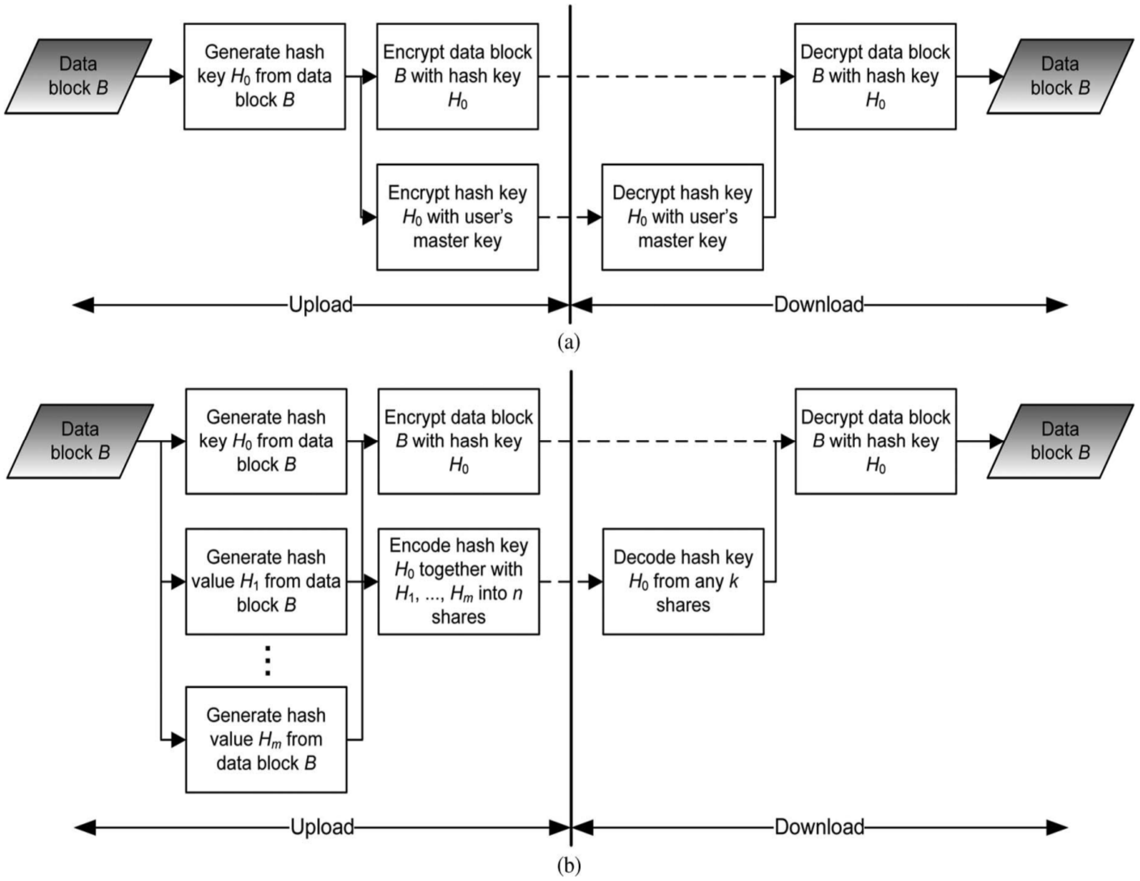
Data block B
Generate hash key H_0 from data block B
Encrypt data block B with hash key H_0
Encrypt hash key H_0 with user's master key
Decrypt hash key H_0 with user's master key
Decrypt data block B with hash key H_0
Data block B
Upload
Download
(a)
Data block B
Generate hash key H_0 from data block B
Encrypt data block B with hash key H_0
Generate hash value H_1 from data block B
Encode hash key H_0 together with H_1, ..., H_m into n shares
Decode hash key H_0 from any k shares
Generate hash value H_m from data block B
Decrypt data block B with hash key H_0
Data block B
Upload
Download
(b)

Fig1: Secure deduplication

(a)Flow diagram of Baseline approach (keeping hash key)

(b) Flow diagram of Dekey (keeping hash key with RSSS).

2. Literature Review/Survey

Data deduplication is important for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space to users. They defined the notions used in based paper, review some secure primitives used secure deduplication. Symmetric Encryption, Convergent Encryption, Proofs of Ownership (PoWs), Ramp Secret Sharing, Secure Deduplication. In 1997.M. Bellare, et.al explains that notion of security and scheme for Symmetric encryption in concentrate security framework. They give several differ notion of security and analyse the concrete complexity of reduction among them. Then they provide concrete security analyses of various method of encryption using a block cipher, including two most popular methods, Cipher block chaining and counter Mode.They had defined two goal first is to study the notion of security for symmetric encryption in the framework for concrete security. That means they looked at the concrete complexity reduction between different notion. To prove the upper and lower bounds so that they can establish tight relationship between notion and compare the stronger and weaker notion. Second one is that to provide concrete security analysis of some specific symmetric encryption schemes. That scheme considers as a pervasive use and yet received any formal analysis in traditional security [1]. In 2002 John R. Douceur et al. explain mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Their mechanism includes First one convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, and second one SALAD, a Self- Arranging, Lossy, Associative Database for aggregating file content and location information in a decentralized, scalable, fault-tolerant manner. Addresses the problems of identifying and coalescing identical files in the Farsite [3] distributed file system, for the purpose of reclaiming storage space consumed by incidentally redundant content. Farsite is a secure, scalable, server less file system that logically functions as a centralized file server but that is physically distributed among a networked collection of desktop workstations. In 2008 M.W.Storer et.al developed two models for secure deduplicated storage authenticated and anonymous. These two designs demonstrate that security can be combined with deduplication in a way that provides a diverse range of security characteristics. In the models they present, security provided through the use of convergent encryption. This technique, first introduced in the context of the Farsite system [4, 5], provides a deterministic way of generating an encryption key, such that two different users can encrypt data to the same cipher text. In both the authenticated and anonymous models, a map is created for each file that describes how to reconstruct a file from chunks. This file is itself encrypted using a unique key. In the authenticated model, sharing of this key is managed through the use of asymmetric key pairs. In the anonymous model, storage is immutable, and file sharing is conducted by sharing the map key offline and creating a map reference for each authorized user. Finally, they examined the information leaks resulting from key compromises and found that the most severe security breaches result from the loss of the client's key. The damage in the event of such a key loss is confined, however, to the user's files. Moreover, the breach of client's keys is a serious threat in most secure systems. While the models they had presented demonstrate some of the ways that security and deduplication can coexist, works remains to create a fully realized, secure, space efficient storage system. Open areas for exploration exist in both security, as well as deduplication that is only their limitation [6]. In 2009 A. yun et.al [7] discussed the introduction of highly parallel overlays of VMs to simplify the management of a cloud infrastructure and increase the overall robustness. To this end, Vinci introduces overlays that interconnect specialized VMs and where a VM may (i) support some applications (ii) protect either a shared file system or the other VMs.Each overlay supports a distinct cloud community, i.e. a set of users with similar security and reliability requirements. A further overlay manages the cloud infrastructure and its VMs can allocate and migrate the various VMs to achieve the required levels of reliability and security. An area of future research is focused on protection against physical attacks so that the cloud provider and the cloud administrators do not have to be trusted. They also investigating how redundancy can be transparently introduced into an overlay to mask faults of hardware components without migrating a VM. Another critical issue is the adoption of multi-core architecture with virtualization support to minimize the overhead introduced by virtualization and the one due to the large number of VMs. Lastly, the complete integration of Vinci with the trusted computing framework supports the definition of a root-of-trust for the assurance checks of the A-VMs. In 2010 p.anderson et.al present FadeVersion, a secure cloud backup system that serves as a security layer on top of today's cloud storage services. FadeVersion follows the standard version-controlled backup design, which eliminates the storage of redundant data across different versions of backups. On top of this, FadeVersion applies cryptographic protection to data backups. Specifically, it enables fine-grained assured deletion, that is, cloud clients can assuredly delete particular backup versions or files on the cloud and make them permanently inaccessible to anyone, while other versions that share the common data of the deleted versions or files will remain unaffected. They implement a proof-of-concept prototype of FadeVersion and conduct empirical evaluation a top Amazon S3[4]. In 2011 Shai Halevi et.al defined Proof of Ownership: The notion of proof of ownership (PoW) is to solve the problem of using a small hash value as a proxy for the entire file in client-side deduplication [11], where the adversary could use the storage service as a content distribution network. This proof mechanism in PoW provides a solution to protect the security in client-side deduplication. In this way, a client can prove to the server that it indeed has the file. Dekey supports client-side deduplication with PoW to enable 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). As they mentioned above, proofs-of-ownership are closely related to proofs of retrievability (POR) and proofs of data possession (PDP). The two main differences are that (a) proofs of retrievability/data-possession often use a pre-processing step that cannot be used in proofs of ownership, and (b) our security notion is weaker than that of proofs of retrievability. Proof-of-ownership is a protocol in two parts between two players on a joint input F (which is the input file). First the verifier summarizes to itself the input file F and generates a (shorter) verification information v. Later, the prover and verifier engage in an interactive protocol in which the prover has F and the verifier only has v, at the end of which the verifier either accepts or rejects. Hence a proof of- ownership is specified by a summary function S (which could be randomized and takes the input file F and a security parameter), and an interactive two-party protocol. In 2012 M. Bellare et.al explain 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-e_cient secure outsourced storage), a goal currently targeted by numerous cloud-storage providers [12]. They introduce an intriguing new primitive that they call Message-Locked Encryption (MLE). An MLE scheme is a symmetric encryption scheme in which the key used for encryption and decryption is itself derived from the message. Instances of this primitive are seeing widespread deployment and application for the purpose of secure deduplication [13], [14], [15], but in the absence of a theoretical treatment, they have no precise indication of what these methods do or do not accomplish. They provide definitions of privacy and integrity peculiar to this domain. Now having created a clear, strong target for designs, they make contributions that may broadly be divided into two parts: (i) practical and (ii) theoretical. In the _rest category they analyses existing schemes and new variants, breaking some and justifying others with proofs in the random-oracle-model (ROM) [16]. In the second category they address the challenging question of ending a standard-model MLE scheme, making connections with deterministic public-key encryption, correlated-input-secure hash functions and locally-computable extractors [17] to provide schemes exhibiting different trade between assumptions made and the message distributions for which security is proven. From our treatment MLE emerges as a primitive that be combines practical impact with theoretical depth and challenges, making it well worthy of further study and a place in the cryptographic pantheon. In 2013 Jin Li,et.al explain for deduplication to protect the confidentiality of sensitive data while supporting deduplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To better protect data security, this paper makes the first attempt to formally address the problem of authorized data deduplication. Different from traditional deduplication systems, the differential privileges of users are further considered in duplicate check besides the data itself. They also present several new deduplication constructions supporting authorized duplicate check in a hybrid cloud architecture. Security analysis demonstrates that our scheme is secure in terms of the definitions specified in the proposed security model. As a proof of concept, they implement a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments using our prototype[22]. They show that their proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations. In 2014 I.Sudha et.al proposed a completely different approach to secure the cloud with the decoy information technology and is called as "Fog Computing". [23]They use this technology to instigate disinformation attacks against malicious insiders, which helps to prevent and distinguish the real sensitive customer data from fake worthless data. The Decoy Information Technology is used for validating whether data access is authorized when abnormal information access is detected. It helps in confusing the attacker with bogus information [24]. In 2014 Jin Li, et.al explain [25] propose Dekey, an efficient and reliable convergent key management scheme for secure deduplication. Dekey applies deduplication among convergent keys and distributes convergent key shares across multiple key servers, while preserving semantic security of convergent keys and confidentiality of outsourced data. They implement Dekey using the Ramp secret sharing scheme and demonstrate that it incurs small encoding/decoding overhead compared to the network transmission overhead in the regular upload/download operations.

3. Proposed work & Objectives:

The basic idea in this paper is that we can eliminate duplicate copies of storage data and limit the damage of stolen data if we decrease the value of that stolen information to the attacker. This paper makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. We propose for providing security in insider attacker as well as outsider attacker and monitoring them we use for that Dekey, user behaviour profiling and Decoy Technology. Dekey is a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments we propose a new construction called Dekey, which provides efficiency and reliability guarantees for convergent key management on both user and cloud storage sides. A new construction Dekey is proposed to provide efficient and reliable convergent key management through convergent key Deduplication and secret sharing. Dekey supports both file-level and block level Deduplication. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. In particular, Dekey remains secure even the adversary controls a limited number of key servers. We implement Dekey using the Ramp secret sharing scheme that enables the key management to adapt to different reliability and confidentiality levels. Our evaluation demonstrates that Dekey incurs limited overhead in normal upload/download operations in realistic cloud environments.

4. Acknowledgement

I owe my deep gratitude to my respected guide Prof. Mr. S.S. KULKARNI who gives me the valuable guidelines with a touch of inspiration and motivation to progress my way through quite substantial barrier between early problem statement and something that resembled a fine work.

5. Conclusion:

The basic idea is that we can limit the damage of stolen data if we decrease the value of that stolen information to the attacker. We can achieve this through a 'preventive' disinformation attack. We posit that secure deduplication services can be implement given additional security features insider attacker on Deduplication and outsider attacker by using the detection of masquerade activity. The confusion of the attacker and the additional costs incurred to distinguish real from bogus information, and the deterrence effect which, although hard to measure, plays a significant role in preventing masquerade activity by risk-averse attackers. We posit that the combination of these security features will provide unprecedented levels of security for the deduplication.

References

[1] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation. Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE, 1997.

[2] 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 Proc. ICDCS, 2002, pp. 617-624.

[3] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer, "Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs", SIGMETRICS 2000, ACM, 2000, pp. 34-43.

[4] A. Adya, W. J. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, Dec.2002. USENIX.

[5] 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 Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS ’02), pages 617–624, Vienna, Austria, July 2002.

[6] M.W. Storer, K. Greenan, D.D.E. Long, and E.L. Miller, ‘‘Secure Data Deduplication,’’ in Proc. StorageSS, 2008, pp. 1-10.

[7] A. Juels and B. S. Kaliski, Jr. Pors: proofs of retrievability for large files. In ACM CCS ’07, pages 584–597. ACM, 2007

[8] H. Shacham and B. Waters. Compact proofs of retrievability. In ASIACRYPT ’08, pages 90–107. Springer-Verlag, 2008.

[9] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. Enabling public verifiability and data dynamics for storage security in cloud computing. In ESORICS’09, pages 355–370. Springer-Verlag, 2009.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In ACM CCS ’07, pages 598–609. ACM, 2007.

[11] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In ACM CCS ’07, pages 598–609. ACM, 2007

[12] P. Anderson and L. Zhang, ‘‘Fast and Secure Laptop Backupswith Encrypted De-Duplication,’’ in Proc. USENIX LISA, 2010,pp. 1-8.

[13] M. Bellare, S. Keelveedhi, and T. Ristenpart, ‘‘Message-Locked Encryption and Secure Deduplication,’’ in Proc. IACR Cryptology ePrint Archive, 2012, pp. 296-3122012:631.

[14] Bitcasa, ini_nite storage. http://www.bitcasa.com/. (Cited on page 3.)

[15] Ciphertite data backup. http://www.ciphertite.com/. (Cited on page 3.)

[16] A. Rahumed, H. Chen, Y. Tang, P. Lee, and J. Lui. A secure cloud backup system with assured deletion andversion control. In Parallel Processing Workshops (ICPPW), 2011 40th International Conference on, pages160-167 IEEE, 2011.

[17] Z. Wilcox-O'Hearn and B. Warner. Tahoe: The least-authority _lesystem. In Proceedings of the 4th ACM international workshop on Storage security and survivability, pages 21-26. ACM, 2008.

[18] S. P. Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. In D. Boneh, editor, CRYPTO 2003, volume 2729 of LNCS, pages 61-77. Springer, Aug. 2003.

[19] A. Yun, C. Shi, and Y. Kim, ‘‘On Protecting Integrity and Confidentiality of Cryptographic File System for Outsourced Storage,’’ in Proc. ACM CCSW, Nov. 2009, pp. 67-76.

[20] M. Ben-Salem and S. J. Stolfo, “Modeling user search-behavior for masquerade detection,” in Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection . Heidelberg: Springer, September 2011, pp. 1–20.

[21] J. Pepitone, “Dropbox’s password nightmare highlights cloud risks,” June 2011.

[22] Salvatore J. Stolfo, Malek Ben Salem and Angelos D. Keromytis “Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud” IEEE Symposium On Security And Privacy Workshop (SPW) YEAR 2012

[23] .Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou” A Hybrid Cloud Approach for Secure Authorized Deduplication” IEEE Transactions On Parallel And Distributed System VOL:PP NO:99 YEAR 2013.

[24] I.Sudha1, A.Kannaki2, S.Jeevidha3” Alleviating Internal Data Theft Attacks by Decoy Technology in Cloud”, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.3, March- 2014, pg. 217-222. B. M. Bowen and S. Hershkop, “Decoy Document Distributor: http://sneakers.cs.columbia.edu/ids/fog/,” 2009. [Online]. Available: http://sneakers.cs.columbia.edu/ids/FOG/

[25] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou “Secure Deduplication with Efficient and Reliable Convergent Key Management” IEEE Transactions On Parallel And Distributed Systems, VOL. 25, NO. 6, JUNE 2014.