Efficient Detection of Duplicate Data using Progressive Techniques

M.Anusuya[#1], R.Kiruba Kumari[*2]

[#1]*M.Phil Research scholar, Department of Computer Science, Padmavani Arts and Science College for Women, Salem-11*

[*2]*Assistant Professor, Department of Computer Science, Padmavani Arts and Science College for Women, Salem-11*

Abstract

Privacy Preserving Data Mining Systems is to propose local data mining and global data mining. It attempts to benefit of extracting useful information from large volumes of data. Privacy-preserving data mining usually has multiple steps that translate to a three-tiered architecture. Online data collection systems are an example of new applications that threaten individual privacy. Already companies are sharing data mining models to obtain a richer set of data about mutual customers and their buying habits as Data Providers, Data Warehouse Server and Data Mining server. Our goal in investigating privacy preservation issues was to take a systemic view of architectural requirements and design principles and explore possible solutions that would lead to guidelines for building practical privacy preserving Central to the strategy are three protocols that govern privacy disclosure among entities as Data collection protocol, Inference Control Protocol and Information sharing Protocol.

Keywords: *Privacy Preserving, Inference Control Protocol, Data Warehouse, Investigating.*

I. INTRODUCTION

Data deduplication is technique to ignore duplicate copies of data, and it has been used to reduce bandwidth of upload and space of storage.Convergent encryption has been adopted for secure deduplication, a critical issue of making convergent encryption is to manage a huge number of convergent keys. Today's objection of cloud storage services is put on account of the increasing volume of data. Deduplication use convergent encryption to make data management scalable.Various businesses, like start-ups, small and medium businesses (SMBs), are increasingly improvement for outsourcing data and computation to the Cloud. Commercial cloud storage services like Dropbox, Mozy, and Memopal, have been applying deduplication to user data inorder to save cost of maintenance.Based upon user's point of view, data outsourcing increases security and privacy concerns. We should trust third-party cloud providers to properly invoke confidentiality, integrity checking, and access control mechanisms contradictory to any insider and outsider attacks. However, deduplication, while enlightning storage and bandwidth efficiency, is adaptable with Convergent key management. Specifically, traditional encryption is responsible for distinct users to encrypt their data with their own keys. Many proposals have been made to secure remote data in the Cloud by applying 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:

A. *User Behaviour Profiling:*

User profiling is a well known Technique to monitor data access in the cloud and detect abnormal data access patterns.This can be applied here to how much a user accesses their information in the Cloud. Such 'normal user' behavior can be used to check whether abnormal access to a user's information is happening. The method of user behavior profiling is commonly used in fraud detection applications. That profiles would naturally include volumetric information, how many documents are typically read and how often.

B. *Decoys*:

Decoy used to monitor data access in the cloud and detect abnormal data access patterns. We start a disinformation attack by responding huge amounts of decoy information to the attacker. This preserve against the misuse of the user's original data. Start disinformation attacks against cruel insiders, preventing them from making the original sensitive users data from fake worthless decoy technology has been used. Its mainly used to serve two purposes:

(1) Authorize data access , when irregular information access is detected, and
(2) Confusing the attacker with bogus information

The decoys, then, serve two purposes:

- Validates whether data access is authorized .
- Confusing the attacker with bogus information.

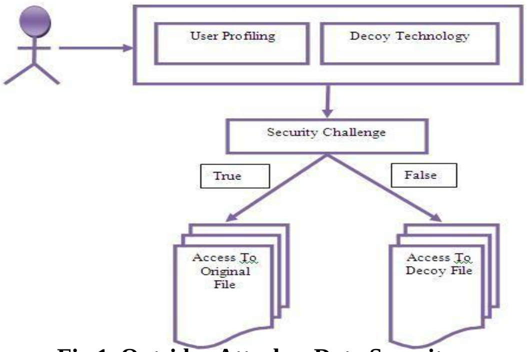

Fig 1. Outsider Attacker Data Security

Outsider attacker protection has to be done by increase the insider attacker with secure deduplication. Convergent encryption provides a data confidentiality in deduplication. It encrypts/decrypts a content with a convergent key, which has to be derived by computing the hash value of the data copy itself [8]. After generating the key data has to be encrypt and then users has to retain the keys and send the cipher text to the cloud. Encryption is deterministic but isame data copies will generate the same convergent key and the same cipher text. It allows the cloud to perform the deduplication on the cipher texts. The cipher text scan is decrypted by the respective data owners with their convergent keys.The new construction is Dekey in which users no need to maintain any keys on their own but instead securely share **out** the convergent key across multiple servers for insider attacker. Dekey uses the Ramp secret sharing scheme and explain that Dekey has limited overhead in realistic environments.

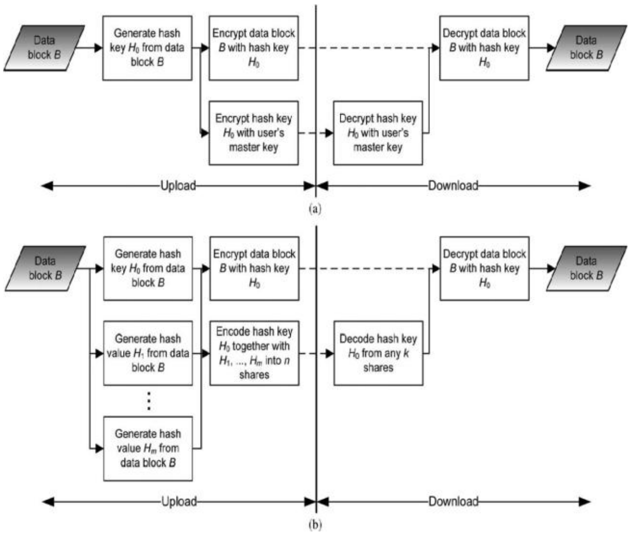
Data block B
Generate hash key H_0 from data block B
Encrypt data block B with hash key H_0
Encrypt hash key H_0 with user's master key
Decrypt hash key H_0 with user's master key
Decrypt data block B with hash key H_0
Data block B
Upload
Download
(a)
Data block B
Generate hash key H_0 from data block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

Fig 2. a) Baseline Approach b)DEKEY

II. LITERATURE 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.

A. Symmetric Encryption

Symmetric encryption is security framework.In traditional encryption different users to encrypt their data with their own keys. A type of encryption where the same key is used to encrypt and decrypt the message. This differs from asymmetric (or public-key) encryption, which uses one key to encrypt a message and another one to decrypt the message. Symmetric encryption is the oldest and best-known technique. A secret key, which can be a number, a word, or just a string of random letters, is applied to the text of a message to change the content in a particular way. This might be as simple as shifting each letter by a number of places in the alphabet. As long as both sender and recipient know the secret key, they can encrypt and decrypt all messages that use this key.The same data content of different users produce different ciphertexts so it is impossible for deduplication .

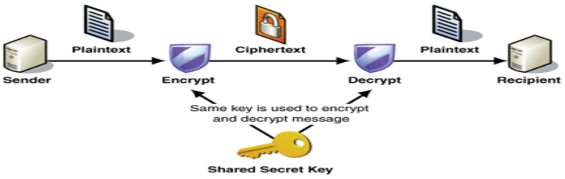

Fig 3.Symmetric Encryption

B. Convergent Encryption

Convergent encryption allows cloud storage services to deduplicate data, without the service having access to the encryption keys used to protect customer files. It provides better privacy than traditional cloud storage. Normally, when cloud services encrypt data, they use their own encryption key. With convergent encryption, the encryption key is derived from the file itself. As such, it produces identical ciphertext from identical plaintext files. Convergent encryption lets cloud storage providers store large amounts of data at low prices, while offering better privacy than traditional cloud storage.The drawback is Privacy concerns have been raised with cloud storage services deduplicating data via convergent encryption. This is because deduplication can be used to "discover" which users are storing a file, if the attacker also has a copy of the file. For instance, an oppressive government could find out which users are storing copies of banned books. Or it could be used to discover which users are storing copyrighted material. This assumes direct access to the servers is given to the outside party.

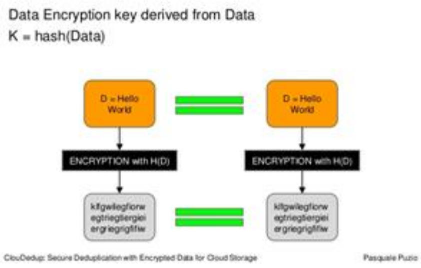
Data Encryption key derived from Data
K = hash(Data)
D = Hello World
D = Hello World
ENCRYPTION with H(D)
ENCRYPTION with H(D)
kifgwiegfiorw egtriegtiergiei ergriegrigfifiw
kifgwiegfiorw egtriegtiergiei ergriegrigfifiw
ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage
Pasquale Puzio

Fig 4.Convergent Encrption

Assuming the attacker has access to the associative array and knows the functions used, what can he do?

The associative array only stores ciphertext X′ and the hashes thereof, H=HB(X′). It never stores the plaintext X or even the hash of the plaintext K.

It is not possible for an attacker to manipulate the stored information. If the attacker where to change a pair (H,X′) into (H,X″) it can be detected because the relation H=HB(X′) will no longer hold. This check only requires knowledge of HB, the verifier does not even have to have the key to the data K. It is therefore advisable to implement this check at any place where such manipulation might occur.

It is possible for an attacker to remove information. If the attacker somehow knows the H it can remove the pair and make the information unavailable. Or it can simply remove the entire associative array. Or the system containing the associative array could catch fire. Fending off these attacks is about having backup systems in place and preventing unauthorized access. The inherent resilience of hash-based content-addressable storage will solve the rest.

1) Confirmation attack

A more fundamental problem with convergent encryption is the confirmation attack. Here an attacker can check if a given key H is in the associative array. If the attacker can do this, he can also check if a given plaintext X is in the associative array by checking the presence of

H=HB(E(HA(X),X)))

If no preventative measures are taken, this could allow an attacker to confirm if the user is in possession of a certain file, for example a banned book or a pirated movie.

2) Offline Brute-Force Attack

In an **off**line attack the attacker can try key combinations at his leisure without the risk of discovery or interference. It could for example mount a brute-force attack by trying out all possible keys until the right one is found.

Determining when the right key is found is hard in conventional encryption systems. Every possible key will result in some sort of plaintext, correct or not. Determining what the correct plaintext looks like involves a heuristic. In general it is impossible to find such a heuristic, and this is what provides the one-time pad with its perfect security.

In convergent encryption it is easy to recognize the correct key. The correct key K will satisfy the equation

K=HA(D(K,X′))

While theoretically interesting, offline brute-force attacks on conventional symmetric cyphers are already possible in practice. Plaintexts often contain easily recognizable structures such as file headers. This can then be used as an effective heuristic to check if the correct key is found. Such an attack will work on any cipher where keys are significantly shorter than the messages, which in practice means anything but the one-time pad.

C. Proof of Owenership

The idea 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 , the could use the storage service as a content distribution network. This proof mechanism in PoW presents a solution to protect the security in client-side deduplication. Like this way, a client can prove to the server that it really has the file. Dekey supports client-side deduplication with PoW to allow users to prove their ownership of data copies to the storage server. Particularly, PoW is implemented as an common algorithm (denoted by PoW) run by the prover (i.e., user) and a verifier (i.e., storage server). The verifier has to derive a short value from a data copy M.In order to prove the ownership of the data copy M, the prover wants to send short value and run a proof algorithm with the verifier. It is passed if and only if short value and the proof is correct. The notations of PoW_F and PoW_B to denote PoW for a file F and block B, respectively. Especially, the notation of $PoW_{F;j}$ will be used to denote a PoW protocol with respect to $T_j(F)$_ $TagGen_{CE}(F; j)$.

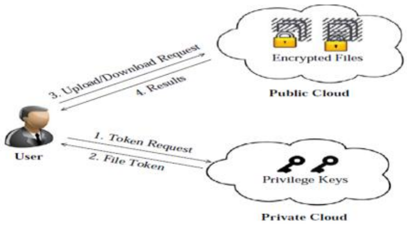

Fig 5.Proof of the Ownership

D. Message Locked Encryption

In 2012 M. Bellare et.al explain formalize a new cryptographic primitive, Message-Locked Encryption (MLE), where the key used for both encryption and decryption are performed is itself derived from the message. MLE presents a way to succeed secure deduplication (space-e_cient secure outsourced storage), a goal currently targeted by numerous cloud-storage providers . 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, 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) . 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 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.

E. Authorized Duplicate Check

In 2013 Jin Li,et.al explain for deduplication to protect the confidentiality of sensitive data while handling deduplication, the convergent encryption technique has been presented to encrypt the data before outsourcing. To protect data security, this paper presents the first attempt to functionally address the problem of authorized data deduplication. Apart from normal deduplication systems, the differential privileges of users are then considered in duplicate check besides the data itself. They also provide several new deduplication constructions supporting authorized duplicate check in a hybrid cloud environment. Security analysis explains 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 plan test bed experiments using our prototype. They show that their proposed authorized duplicate check scheme obtains minimal overhead compared to normal operations. In 2014 I.Sudha et.al proposed a completely distinct approach to secure the cloud with the decoy information technology and is called as “Fog Computing” . They use this technology to activate disinformation attacks against malicious insiders, which helps to prevent and distinguish the real perceptive customer data from fake worthless data. The Decoy Information Technology is used for validating whether data access is authorized even when abnormal information access is detected. It helps in confusing the attacker with bogus information. In 2014 Jin Li, et.al explain propose Dekey, an efficient and reliable convergent key management scheme for secure deduplication. Dekey supplies deduplication among convergent keys and distributes convergent key shares across multiple key servers, while reserving semantic security of convergent keys and confidentiality of outsourced data. They present Dekey using the Ramp secret sharing scheme and explains that it incurs small encoding/decoding overhead related to the network transmission overhead in the regular upload/download operations.

III. PERSONALIZED RECOMMENDATION ALGORITHM

The notion in this paper is that we can avoid duplicate copies of storage data and limit the damage of stolen data by decreasing the value of that stolen information to the attacker. This paper comples the first attempt to address the problem of succeeding efficient and reliable key management in secure deduplication. We propose for providing security in both insider attacker as well as outsider attacker and also monitoring them we use for that Dekey, user behaviour profiling and Decoy Technology. Dekey is a new construction in which users no need to maintain any keys on their own but they securely distribute the convergent key shares over multiple servers. Dekey using the Ramp secret sharing scheme and explains that Dekey incurs limited overhead in realistic environments. To propose a new construction called Dekey, which provides efficiency and reliability promises for convergent key management on both user and cloud storage sides. Dekey is introduced to present efficient and reliable convergent key management through convergent key Deduplication and secret sharing. Dekey handles both file-level and block level Deduplication. Security analysis explains that Dekey is secure in terms of the definitions states in the proposed security model. Specifically, Dekey remains secure even the adversary handles a limited number of key servers. Dekey uses the Ramp secret sharing scheme that enables the key management to prepare to different reliability and confidentiality levels. Our evaluation demonstrates that Dekey incurs limited overhead in bothupload/download operations in realistic cloud environments.

IV. CONCLUSION

The basic idea in this paper is to limit the damage of stolen data if we decrease the value of that stolen information to the attacker. To succed this a 'preventive' disinformation attack has to be used. We present that secure deduplication services can be present with 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 imitate activity by risk-averse attackers. We posit that the combination of these security features will provide unparalleled 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

[6] International Conference on Distributed Computing Systems

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

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

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

[10] 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.

[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] 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

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

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

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

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] J. Pepitone, "Dropbox's password nightmare highlights cloud risks," June 2011.

[23] 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

[24] .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.

[25] 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/

[26] 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.