

Secure and Public Cloud Storage Auditing with Deduplication Environment

JS. Abishek Rosario^{#1}, B. Ganeshkrishna^{#2}, R.Bhuvaneshwar^{#3}, A. Bhuvaneshwaran^{*4}

[#] CSE Department, Kalasalingam Academy of Research and Education, Krishnankoil, Srivilliputhur, Tamil Nadu

^{*} Professor in Computer Science and Engineering at the Kalasalingam Academy of Research and Education, Tamil Nadu, India

Abstract— The use of cloud computing has increased rapidly in many organizations. Cloud computing provides many benefits in terms of low cost and accessibility of data. Ensuring the security of cloud computing is a major factor in the cloud computing environment, as users often store sensitive information with cloud storage providers but these providers may be untrusted. Dealing with “single cloud” providers is predicted to become less popular with customers due to risks of service availability failure and the possibility of malicious insiders in the single cloud. A movement towards “multi-clouds”, or in other words, “interclouds” or “cloud-of clouds” has emerged recently. DNA based encryption for security is the trend. DNA cryptography was a breakthrough in the field of security which uses bio-molecular concepts and gives us new hope of unbreakable algorithms but the concepts need to be exploited more especially in the cloud computing. This paper discusses cloud computing features, service models, and security issues and proposes a DNA based encryption algorithm for securing data in cloud environment which will be cost effective and secure by using bio-computational techniques. The suggested algorithm uses indexing and DNA steganography techniques along with binary coding rules which make algorithm secure as it is an additional layer of biosecurity than conventional cryptographic techniques.

Index Terms— Cloud computing, Data security issues, BiDirectional DNA Encryption Algorithm

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, deduplication has been a well-known technique and has attracted more and more attention recently. Data deduplication 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, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take place at either the file level or the block level. For file level deduplication, it eliminates duplicate copies of the same file. Deduplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data deduplication brings a lot of benefits, security and privacy concerns arise as users’ sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication. 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 deduplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making deduplication 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. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. However, previous deduplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized deduplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and his own privileges as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege

stored in cloud. For example, in a company, many different privileges will be assigned to employees. In order to save cost and efficiently management, the data will be moved to the storage server provider (SCSP) in the public cloud with specified privileges and the deduplication technique will be applied to store only one copy of the same file. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with specified privileges to realize the access control.

II. PROBLEM FORMULATION

The research challenge is how to make correct proactive real-time defense decision during an earlier stage of attack with less harm (Liu et al., 2005) and consuming minimal system resource?

III. PROPOSED METHODOLOGY

A. Hybrid Architecture for Secure Deduplication

At a high level, our setting of interest is an enterprise network, consisting of a group of affiliated clients (for example, employees of a company) who will use the S-CSP and store data with deduplication technique. In this setting, deduplication can be frequently used in these settings for data backup and disaster recovery applications while greatly reducing storage space. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions. There are three entities defined in our system, that is, users, private cloud and S-CSP in public cloud. The access right to a file is defined based on a set of privileges. The exact definition of a privilege varies across applications. For example, we may define a role based privilege according to job positions (e.g., Director, Project Lead, and Engineer), or we may define a time-based privilege that specifies a valid time period (e.g., 2014-01-01 to 2014-01-31) within which a file can be accessed. A user, say Alice, may be assigned two privileges “Director” and “access right valid on 2014-01-01”, so that she can access any file whose access role is “Director” and accessible time period covers 2014-01-01. Each privilege is represented in the form of a short message called token. Each file is associated with some file tokens, which denote the tag with specified privileges. A user computes and sends duplicate-check tokens to the public cloud for authorized duplicate check. Users have access to the private cloud server, a semi trusted third party which will aid in performing deduplicable encryption by generating file tokens for the requesting users. We will explain further the role of the private cloud server below. Users are also provisioned with per-user encryption keys and credentials (e.g., user certificates). In this paper, we will only consider the file level deduplication for simplicity. In another word, we refer a data copy to be a whole file and file-level deduplication which eliminates the storage of any redundant files. Actually, block-level deduplication can be easily deduced from file-level deduplication, which is similar to. Specifically, to upload a file, a user first performs the file-level duplicate

check. If the file is a duplicate, then all its blocks must be duplicates as well; otherwise, the user further performs the block-level duplicate check and identifies the unique blocks to be uploaded. Each data copy (i.e., a file or a block) is associated with a token for the duplicate check. This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcing service and stores data on behalf of the users. To reduce the storage cost, the S-CSP eliminates the storage of redundant data via deduplication and keeps only unique data. We assume that S-CSP is always online and has abundant storage capacity and computation power. A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. In the authorized deduplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges. Compared with the traditional deduplication architecture in cloud computing, this is a new entity introduced for facilitating user’s secure usage of cloud service. Specifically, since the computing resources at data user/owner side are restricted and the public cloud is not fully trusted in practice, private cloud is able to provide data user/owner with an execution environment and infrastructure working as an interface between user and the public cloud. The private keys for the privileges are managed by the private cloud, who answers 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.

B. DNA

DNA has an excellent storage capacity of 106 TB in 1gm of DNA, which indicates a few grams of DNA may have the capacity to store whole data available in the world [12]. Cloud computing techniques also provide data storage so instead of using DNA for data storage exploiting DNA based encryption for data stored in the cloud will be a combination of trending techniques which ensures data security, Confidentiality, integrity and authentication are considered as most important aspects of information security whereas context is also an important aspect depending on it, the level required for data security will be different. Cryptographic techniques have a major role in securing data and depending on the context different encryption techniques can be used. The encryption techniques are broadly classified as symmetric key cryptography and asymmetric key cryptography depending on the context it can be used in conjunction with DNA computing.

IV. VALIDATION OF RESULTS

A. File Size

To evaluate the effect of file size to the time spent on different steps, we upload 100 unique files (i.e., without any deduplication opportunity) of particular file size and record

the time break down. Using the unique files enables us to evaluate the worst-case scenario where we have to upload all file data. The average time of the steps from test sets of different file size. The time spent on tagging, encryption, upload increases linearly with the file size, since these operations involve the actual file data and incur file I/O with the whole file. In contrast, other steps such as token generation and duplicate check only use the file metadata for computation and therefore the time spent remains constant. With the file size increasing from 10MB to 400MB, the overhead of the proposed authorization steps decreases from 14.9% to 0.483%.

B. Number of Stored Files

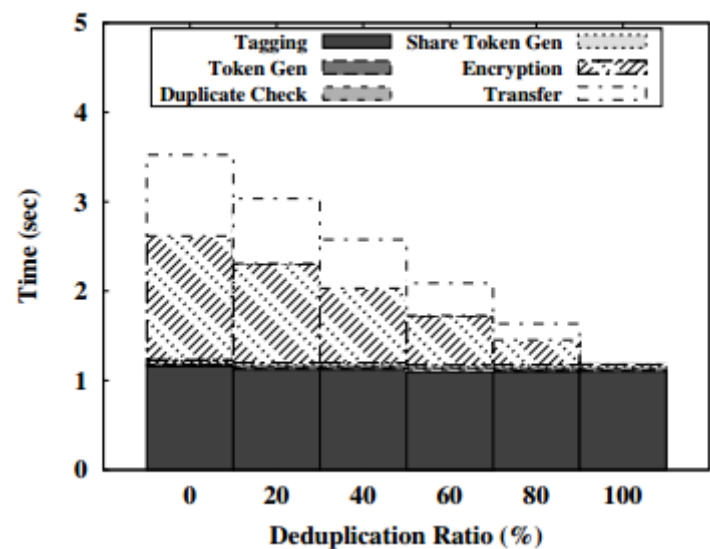
To evaluate the effect of number of stored files in the system, we upload 10000 10MB unique files to the system and record the breakdown for every file upload. Every step remains constant along the time. Token checking is done with a hash table and a linear search would be carried out in case of collision. Despite of the possibility of a linear search, the time taken in duplicate check remains stable due to the low collision probability.

C. Deduplication Ratio

To evaluate the effect of the deduplication ratio, we prepare two unique data sets, each of which consists of 50 100MB files. We first upload the first set as an initial upload. For the second upload, we pick a portion of 50 files, according to the given deduplication ratio, from the initial set as duplicate files and remaining files from the second set as unique files. The average time of uploading the second set. As uploading and encryption would be skipped in case of duplicate files, the time spent on both of them decreases with increasing deduplication ratio. The time spent on duplicate check also decreases as the searching would be ended when duplicate is found. Total time spent on uploading the file with deduplication ratio at 100% is only 33.5% with unique files.

D. Privilege Set Size

To evaluate the effect of privilege set size, we upload 100 10MB unique files with different size of the data owner and target share privilege set size. The time taken in token generation increases linearly as more keys are associated with the file and also the duplicate check time. While the number of keys increases 100 times from 1000 to 100000, the total time spent only increases to 3.81 times and it is noted that the file size of the experiment is set at a small level (10MB), the effect would become less significant in case of larger files.



V. CONCLUSIONS

In this journal, the notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new deduplication 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] 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 Security Symposium, 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] GNU Libmicrohttpd. <http://www.gnu.org/software/libmicrohttpd/>.
- [11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.
- [12] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.

- [13] libcurl. <http://curl.haxx.se/libcurl/>.
- [14] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In Proc. of APSYS, Apr 2013.
- [15] W. K. Ng, Y. Wen, and H. Zhu. Private data deduplication protocols in cloud storage. In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.
- [16] R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and Communications Security, pages 81–82. ACM, 2012.
- [17] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In Proc. USENIX FAST, Jan 2002.
- [18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In 3rd International Workshop on Security in Cloud Computing, 2011.
- [19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29:38–47, Feb 1996.
- [20] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. In Technical Report, 2013.
- [21] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller. Secure data deduplication. In Proc. of StorageSS, 2008.
- [22] Z. Wilcox-O’Hearn and B. Warner. Tahoe: the least-authority filesystem. In Proc. of ACM StorageSS, 2008.
- [23] J. Xu, E.-C. Chang, and J. Zhou. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In ASIACCS, pages 195–206, 2013.
- [24] J. Yuan and S. Yu. Secure and constant cost public cloud storage auditing with deduplication. IACR Cryptology ePrint Archive, 2013:149, 2013.
- [25] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan. Sedic: privacyaware data intensive computing on hybrid clouds. In Proceedings of the 18th ACM conference on Computer and communications security, CCS’11, pages 515–526, New York, NY, USA, 2011. ACM.