EFFICIENT BLOCK-WISE IMAGE COMPARISON AND STORAGE REDUCTION USING DICE PROTOCOL

Santhosh Kumari[1], Archana[2], Kavya Shree[3], Ashwini[4], Chitra M[5]
Department of Computer Science , Dr T Thimmaiah Institute of Technology [1]santhoshkumariv@gmail.com,[2]archana.ramu2015@gmail.com, [3]kavyashreen966@gmail.com, [4]ashwininataraj1298@gmail.com, [5]chitramr123@gmail.com

Abstract

In this paper we introduce a secure deduplication scheme for near identical (NI) images using the Dual Integrity Convergent Encryption (DICE) protocol. An image is divided into blocks and the DICE protocol is applied on each block separately instead of applying on the entire image. As a result, the blocks that are similar between two or more NI images are deposited only once at the cloud. This paper provides a technique to accomplish secure image deduplication at the block level based on the DICE protocol which exhibits that the greater the resemblance of the images, the smaller the number of blocks deposited at the cloud. Applying secured data deduplication to such data files could remarkably minimize the cost and space required for their storage.

Index Terms: Security;Image Deduplication; Cloud Storage.

I.INTRODUCTION

Cloud computing provides users with the platform to profit cloud services on demand which include primarily storage, database, networking, and software services over the Internet. Whether a user is watching movies, listening to audio, taking pictures, hosting websites or creating new apps, cloud computing is an essential part of all these services. Cloud service providers (CSPs) charge their users a nominal fee for the use of these services. Therefore, it is important for the CSPs to maintain a tradeoff between the cost of the services they provide and the fees that they charge to their users, as maintaining and storing the huge volume of users' data, along with the bandwidth usage incur costs for the CSPs.

Cloud service providers (CSPs)depends on deduplication techniques for eliminating duplicate data and thus minimize bandwidth and storage requirements. However, it is equally important for CSPs to ensure the privacy and security of users' data. To address both these problems, secured data deduplication was established.

Identifying duplicate copies in the encrypted image and video data is a significant challenge. The existing techniques developed for generic data may not be actually suitable for multimedia data.

In this paper, we are using a secure block level image deduplication scheme which removes the near identical images (formally defined in Section 3) in encrypted form, so by protecting the confidentiality of the images. The proposed method make use of the Dual Integrity Convergent.

Encryption (DICE) protocol that the authors proposed in their recent work [3]. Our main idea is to divide the image into blocks and apply the DICE protocol on each block separately. Each block is encrypted using AES with a key that is formed by hashing the image blocks.

This means that identical blocks in any two images will produce the identical cipher text, which allows the CSPs toper form deduplication on the cipher text blocks. The communication and bandwidth requirements are also reduced because only one tag is generated from the cipher text. The security of the scheme has been determined experimentally as well as theoretically.

The rest of the paper is organized as follows: section II discusses the related work in detail. In section III, we describe the proposed method. Next, we present the security and performance

analyses in section IV and section V, respectively. Finally, section VI concludes the paper with a discussion on the future work.

II.RELATED WORK

Here, we discuss related work from two features:

1)Designed cryptographic protocols that are for secure data deduplication, assuming that the data is generic, and 2) modifications of these protocols for secure image data deduplication. From the cryptographic protocols characteristic, Bellare et al. have proposed the Message Locked Encryption (MLE) scheme [4], which defines the state-of-the-art protocol standards. There are several other MLE-based strategies in the literature, such as Convergent Encryption (CE), HCE1, HCE2 and Randomized Convergent Encryption (RCE) [5]. All of the above mentioned strategies provide deduplication along with the necessary cryptographic security characteristics [6], [7].But still, all these strategies are unprotected to security attacks; among them specifically mentioned is the poison attack which includes the duplicate copy replacement attack and the erasure attack, where the malicious user replaces the actual file with the corrupted file which is been modified by him/her. This will make, honest users lose their files and has been made to download the faked ones.

A deduplication strategy could be on the server side or on the client side. In server side deduplication, the client transfer the files (including duplicates) to the server after which the server removes the identicals and stores the unique files correspondingly. Simultaneously, the server verifies the client to retrieve the files and update the metadata. With this, the overhead is higher on the server side and this results in consumption more bandwidth and computation cost. On the other hand, client side deduplication strategy, works such that the computations are done first on the client side it self by producing tags. The server will be sent only the tags as an alternative of the entire file and further communication continues through tag insepction.

(a)

(b)

Figure 1: Two near identical images differing in (a) some blocks (difference is the bird) (b) resolution.

In the client side deduplication strategy, the purpose is to have less overhead at the server. Therefore, the client side deduplication strategy is more efficient, especially as the number of clients gets growing [8]. Both strategies, client side and server side, have their own advantages and disadvantages [9].

Agarwala et al. [3] recently proposed the DICE protocol Which has reduced computations, communication and bandwidth requirements just by incorporating one tag. most of the computations in this protocol were performed on the client side. The deduplication is performed at the file level, where the duplicate files are identified by applying hash functions on the entire file and then check whether the hash values are matching. In the case of image data, applying hash on the entire image data is not efficient, as the hash value may vary even if two images are differing only by one pixel value, there by not achieving the goal of deduplication. Rather, the image files could be broken down into blocks and deduplication can be performed by first applying hash on the blocks, and then for the similarity between the hash values of respective blocks, which is the method implemented in this paper.

A number of researchers has been addressed Secure deduplication of images by. For instance, Gang et al. [10] to perform image deduplication he took the deduplication of the entire image and applied the CE scheme paired

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-6, ISSUE-6, 2019
DOI: 10.21276/ijcesr.2019.6.6.30

with Attribute Based Encryption. In another work, Fatema et al. [11] to perform deduplication used SPIHT compression characteristics and partial encryption along with image hashing. The partial encryption scheme provides security against the CSP and the image hashing technique makes it possible to identify the identical images which are compressed and encrypted for deduplication. In their work, the client applies the image compression algorithm first, then uses the partial encryption scheme and finally computes the image hash signature. The signature is then sent to the CSP to check for deduplication. Another work by Li et al. [12], proposed a scheme called Client-based Security Provable Deduplication of Multimedia Data (CSPD),in which using fuzzy methods it checked the duplication of images . Unlike other strategies that check duplication on the hash values of the image , in their strategy, after the client uploads certain number of specifications of the image to the CSP, the CSP applies Hamming distance based on the specifications of the stored images and searches for identical images in the database. In their work, Xuan Li et al. [13] propose a secure perceptual similarity deduplication scheme where they use the pHash algorithm to determine the similarity of the image hashes stored at the cloud by determining the hamming distances between the hashes stored image. They also make use of a group key, in which all the members in a particular group can upload and download the images by using this group key. All these group keys are kept and employed by the users according to the group of users sharing data. Their scheme is also vigorous to common image

processing operations such as resizing and compression.

Most of the work that we found in the literature is based on computing the hash of the entire image at once, instead of converting the images into blocks. Such schemes are mainly useful for identical images, however they are not suitable for nearly identical images. We believe that breaking down the images into smaller structures like blocks, considering the parameters of the blocks and then performing deduplication at the block level will significantly help to achieve a more accurate method for deduplication.

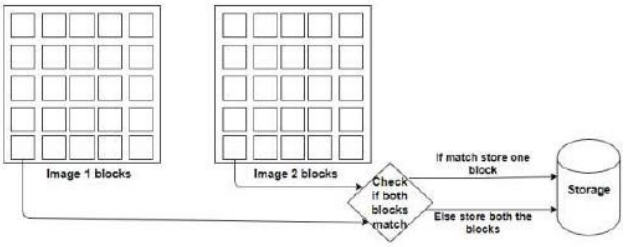

Figure 2: Block level deduplication of two images.

III.PROPOSED WORK

A.Near Identical Images

In this work, we interpret the near identical image scenario as two or more images which have the same background, but there is either a change in a specific block, or some of the pixels are different. These differing pixels may be concentrated in a specific region or may be dispersed all around the image (two examples of near identical images are shown in Figure 1). Below, we provide a conventional definition of near identical images.

Definition 1: *δ-NI Images: An image pair (I,I) is called δ-NI (or δ-near identical), if the ratio of blocks that aresame in I and Iis δ(I,I) ∈[0,1].*

Here, the images are nearly identical content-wise (for example, pictures of two different persons taken in the same pose, background, setup, etc.). For the secure deduplication of δ-NI images, we apply block wise translation of the images and check if the two images match block wise, as shown in Figure 2. The size of the blocks can differ (e.g. 4×4, 8×8, 16×16). We match the images by mapping the first block of the image with the first block of the second image and continue to do this until the last block. After we convert the image block wise, we run the DICE protocol on the blocks. For the blocks that match, we maintain only one copy of the block on the cloud storage.

B.Secure Deduplication of NI-images with DICE

In this section we provide a complete view of the secured block level image deduplication

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-6, ISSUE-6, 2019
DOI: 10.21276/ijcesr.2019.6.6.30

strategy based on the DICE protocol. In the following, we first report the system and threat models and examine the assumptions made in the DICE protocol when applied to NI-images, then we present the alteration of the DICE protocol for NI-images. We call this new protocol DICE-NI.

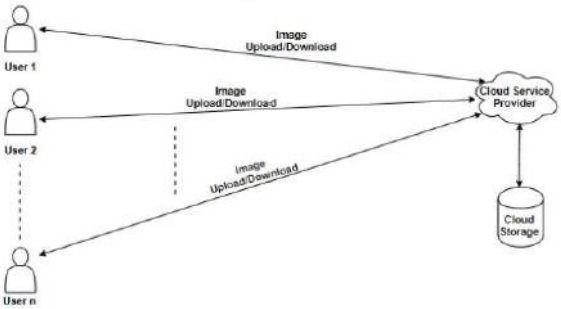

Figure 3: System overview.

1)System model: The system model is comprised of users and the CSP, where there could be multiple users retrieveing the cloud to upload or download images, as can be seen in Figure 3.

CSP: The CSP provides the storage services to the users who have been permited access and are thus authorized to use the cloud services. For the use of these services and the storage and maintenance of their data, users are charged a nominal fee. The CSP must, in turn, ensure the security and privacy of the users' data.

Users: Users are the people who are authorized to access the cloud services. They have the encrypted file, which in this particular scenario is comprised of an image that they upload to the CSP using the DICE protocol. Here, different users can concurrently access the cloud, where they convert their images block wise and upload them to the cloud. The CSP is required to preserve an access file that consists of the blocks, the ID of the image and the user credentials. Before uploading the image block wise, the client checks for the accessible of the blocks at the cloud. If a block exists, then a link is provided to the particular user, otherwise a request to upload the block is sent.

2)Threat model: In this section we examine the threats from the failure of an adversary, where they are attentive in knowing the content of the images. We examine the adversary to be malicious, the CSP to be semi-malicious and the user to be truthful. An adversary could be an interior or an exterior adversary. An interior adversary is more attentive in knowing the content of a file that might belong to the CSP. As a result, we examine the CSP to be semi-honest. Conversely, an exterior adversary is attentive in both the content and the owner of the file. In this scenario, the threats could be generated by straight accessing the cloud services or by gaining access to the channel. If the attacker gains access to the cloud, they could try to delete the content of the file or restore it with a different file. But irrespective of the objective of the adversary, the protocols used by the CSPs should be secure enough to guarantee the detection and prevention of those attacks.

3)Assumptions: We assume that the users have been successfully authenticated by the CSP and granted the necessary rights to access the cloud resources. We also assume that the hash functions are collision resistant, and that the cryptographic primitives being used to design the block level image deduplication protocol are secure and are computationally infeasible for any adversary to break, given sufficient computing resources and power.

4)DICE-NI protocol: The user first divides the image into a fixed number of blocks. Each block size could be of variable length, anywhere from 4×4, 8×8 to 16×16. After converting the image into blocks, the user runs the client portion of the DICE protocol on each block. As per the DICE protocol, the client calculates the key Ki ← H(Bi) where H is the hash function, and Bi is the ith block of the image. Next, the client computes the cipher text Ci as Ci ← E (,) and the tag T as Ti ← H ( ), where E is the encryption strategy.

After computing the keys, the cipher text and the tags, the user obtains the following vector {K11,K12,...Kmn}, {C11,C12,...Cmn}, {T11,T12,...,Tmn} for an image I, where mn is the total number of blocks. At this stage, the user sends the tag vector {T11,T12,...Tmn} to the CSP and checks for its existence in the cloud. The CSP then runs a explore in the tag stock for the existence of the tags from the tag vector and sends a request for only those blocks for which no match was establish. The client then sends the cipher text of those particular blocks to the CSP, who stores them along with the user's credentials and modernize its tag store by computing T ← H (Ci).

At the time of download, the user sends the tag vector and user id, and the CSP explores its tag

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-6, ISSUE-6, 2019
DOI: 10.21276/ijcesr.2019.6.6.30

store to find the corresponding tag and cipher text block as Ti = T'i. If there is a match found, then the correlate cipher text block is sent to the particular user, otherwise the CSP sends an acknowledgement that the image is not found. After the user receives the cipher text, the tag is computed from the received cipher text block as T"i = H(Ci) and is matched with the stored tag as Ti = T"i . If there is a match found, then the decryption process starts as Bi ← D(Ki,C i), where D is the decryption strategy; otherwise the user sends an acknowledgement to the CSP that the block has been corrupted.

The above protocol is based on the MLE scheme. There are other widely used MLE based schemes such as CE, HCE1, HCE2 and RCE, but the reason that we chose to implement DICE based block level deduplication of the images is because DICE is a client based strategy which is secured against the poison attack, unlike HCE1, HCE2 and RCE, which are not secured against the poison attack. While CE is secured against the poison attack, it is a server sided strategy. At the same time, the running time of DICE is considerably less than that of the CE scheme but equivalent to other client based strategies. Agarwala et. al [3] provide more details on the comparison and security analysis of the various MLE based schemes.

IV.SECURITY ANALYSIS

The MLE based scheme reported above conserves the privacy of the data because the hash value of each block is handed-down as the image encryption key for that specific block, then AES is applied as the image encryption strategy. Here, the key is directly intellectual to the hash value of the image block content. As a solution, the key values and their equivalent cipher text are highly implausible to be the same, rendering it hard to discover any relation between them. This, in turn, makes it hard for an adversary to launch a dictionary attack on the block wise encryption strategy. Also the block wise image dedudplication scheme is secure against the poison attack, because the DICE protocol is secure against it. To further certify this point we provide the following theorem.

Theorem 1: Block level image deduplication is secured against poison attack.

Proof: Let us suppose that the adversary computes the forged key, cipher text and tag as , and , respectively. They continue with ×Bi for the block B, i ≤ n where ×Bi = T ×Bi. For the condition to hold true, the adversary has to compute ×B and make it equal to C×Bi. At the same time, the key is directly intellectual to the hash value of the image block content which results in the adversary having to ensure that ×B=K×B. This is computationally infeasible because of the one way property of the hash functions. Hence, the DICE-NI scheme is secure against the poison attack.

V.PERFORMANCE ANALYSIS

We approve the performance of the DICE-NI protocol by handling simulations performed on a network between the client and the server using Java sockets on a Windows 10 platform 64 bit with Intel Core i5. We used MD5(Message digest version 5) to compute the file hash and AES to encrypt and decrypt the content.

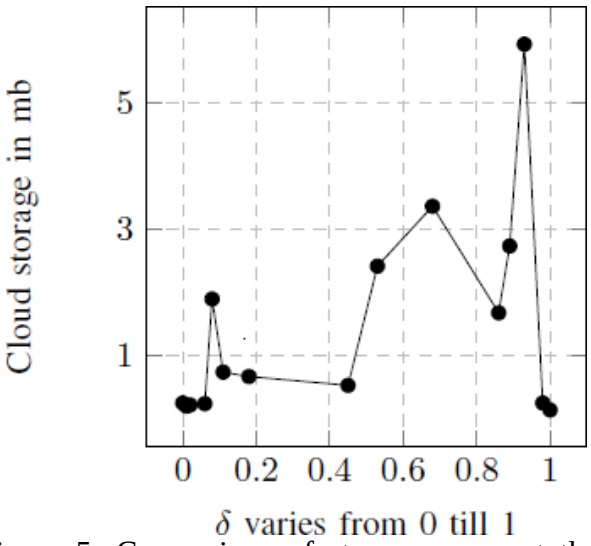

Figure 5: Comparison of storage space at the cloud with an increasing ratio (δ) between image pairs that have similar blocks.

We ran the DICE-NI protocol on a 20 image data set (10 image pairs) where the image size ranged from 100kb to 500kb and we restricted the number of users to one. The image data set comprise images where some have the same resolution but only certain blocks were different, whereas others have different resolution but looked nearly identical with slight variations on the blocks. From Figure 6 we can observe that for different values of δ we see a reduction in the storage space at the cloud.

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-6, ISSUE-6, 2019
DOI: 10.21276/ijcesr.2019.6.6.30

For example, with δ = 0.73, the total storage space taken by two images at the cloud is150kb, whereas the total size of the image pair together is 190kb. Similarly, for δ = 0.67 total storage space demand was 420kb. After storing all of the images block wise, we observed that the total storage space for all 20 images was reduced to 375kb from 420kb, a savings of 45kb. We will observe more deviations when there are large number of NI images transmit by different users concurrently to the CSP.

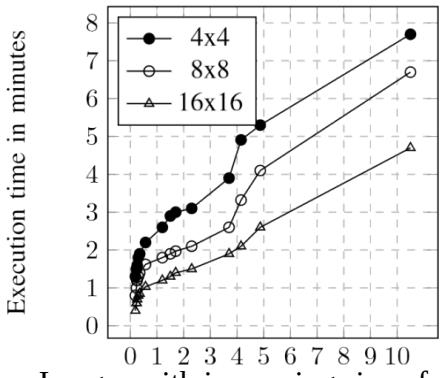

Figure 6: Comparison of DICE-NI execution time with varying number of block size.

In the second trial we run the DICE-NI scheme by changing the block size of the images to check the total time taken to execute the protocol. From Figure 6 we observe the total time taken for DICE-NI execution for images of varying block sizes: 4 × 4, 8 × 8 and 16 × 16.

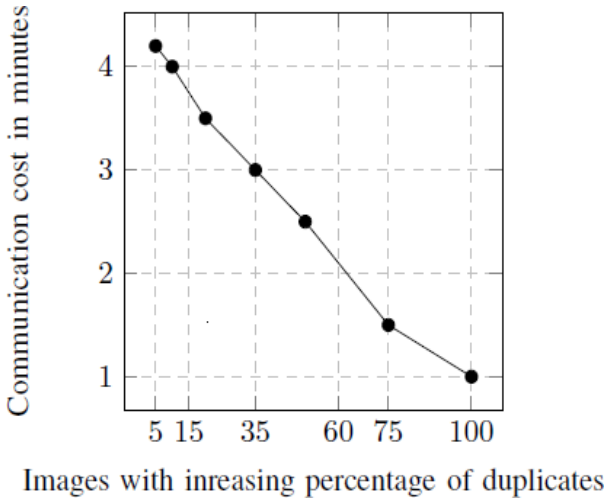

Figure 7: Comparison of communication cost with increasing percentage of NI blocks.

It is crystal clear that the block size is directly analogous to the execution time of DICE-NI; smaller size blocks take more time to execute than the larger blocks. despite larger blocks take less execution time, they may steer to other arms. For example, larger blocks could result in more data being stored at the CSP, or the users may not be in position to recover the complete image properly at the time of download due to the loss of some pixel values at the time of restoring. Even though smaller block sizes could help to achieve better deduplication, the tags and the keys created and deposited at the client side are of fixed size which is regardless of the block size. We do not want the block size to be too small because it may collapses the whole aspire of saving storage space. With a varying block size, however, some nearly identical images attain better deduplication results, while other images do not. We observe that establish an optimal block size with a acceptable value for the threshold δ is a energizing task when running these strategies.

In our third trail we observed the communication time taken by the users to upload and download images at the CSP. From Figure 7 we can perceive that diminish in the total communication time by increasing the percentage of δ, while keeping other factors constant, such as the block size, number of users retrieving the cloud and image size. We would desire to test this promote by varying all the parameters in order to determine an optimal function to lessen the overall cost.

VI.CONCLUSION

In this paper we addresses a method to perform secure image deduplication at the block level based on the DICE protocol. We are providing more security to the cloud data using cryptographic encryption techniques so that we are able to avoid tampering of the cloud data by adversary and maintains its integrity. We found that the greater the similarity of the images, the smaller the number of blocks stored at the cloud.However, the constraint here was that the images were nearly identical with small deviations between them. In the future we would like to address this issue on a broader spectrum where we add more image operations like scaling, rotation, cropping, multiple viewpoints, lighting conditions, and compression with different file formats, and test them at the cloud.

ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-6, ISSUE-6, 2019
DOI: 10.21276/ijcesr.2019.6.6.30

REFERENCES

[1]T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou,―Towards efficient fully randomized message-locked encryption,‖ in The 21st Australasian Conference on Information Security and Privacy, Melbourne, VIC, Australia, 2016, pp. 361–375.

[2]D. Koo, J. Hur, and H. Yoon, ―Secure and efficient deduplication over encrypted data with dynamic updates in cloud storage,‖ in Frontier and Innovation in Future Computing and Communications, Dordrecht, 2014, pp. 229–235.

[3]A. Agarwala, P. Singh, and P. K. Atrey, ―DICE: A dual integrity convergent encryption protocol for client side secure data deduplication,‖ in IEEE International Conference on Systems, Man, and Cybernetics, Banff, Canada, 2017, pp. 2176–2181.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, ―Messagelocked encryption and secure deduplication,‖ in Advances in Cryptology – 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, 2013, pp. 296–312.

[5]M. Bellare and S. Keelveedhi, ―Interactive message-locked encryption and secure deduplication,‖ in Public-Key Cryptography – 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, 2015, pp. 516–538.

[6] J.Stanek,A.Sorniotti,E.Androulaki,andL.Kencl, ―Asecure data deduplication scheme for cloud storage,‖ in Financial Cryptography and Data Security, Berlin, Heidelberg, 2014, pp. 99–118.

[7]M. W. Storer, K. Greenan, D. D. Long, and E. L. Miller, ―Secure data deduplication,‖ in Proceedings of the 4th ACM International Workshop on Storage Security and Survivability, Fairfax, Virginia, USA, 2008, pp. 1–10.

[8]J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, ―Reclaiming space from duplicate files in a serverless distributed file system,‖ in The 22nd International Conference on Distributed Computing Systems, Vienna, Austria,, 2002, pp. 617–624.

[9]K. Keonwoo, Y. Taek-Young, J. Nam-Su, and C. Ku-Young, ―Client-side deduplication to enhance security and reduce communication costs.‖ ETRI Journal, vol. 39, no. 2, pp. 116 – 123, 2017.

[10]H. Gang, H. Yan, and L. Xu, Secure Image Deduplication in Cloud Storage. Cham: Springer International Publishing, 2015, pp. 243–251.

[11]F. Rashid, A. Miri, and I. Woungang, ―Secure image deduplication through image compression,‖ J. Inf. Secur. Appl., vol. 27, no. C, pp. 54–64, 2016.

[12]D. Li, C. Yang, C. Li, Q. Jiang, X. Chen, J. Ma, and J. Ren, ―A client-based secure deduplication of multimedia data,‖ in IEEE International Conference on Communications, Paris, France, 2017, pp. 1–6.

[13]X. Li, J. Li, and F. Huang, ―A secure cloud storage system supporting privacy-preserving fuzzy deduplication,‖ Soft Computing, vol. 20, no. 4, pp. 1437–1448, 2016.