A Client-based Secure Deduplication of Multimedia Data

Danping Li[*1], Chao Yang[*2], Chengzhou Li[§3], Qi Jiang[*4], Xiaofeng Chen[*5], Jianfeng Ma[*6], and Jian Ren[†7]
[*]School of Cyber Engineering, Xidian University, China
[§]School of Electronic Engineering, Xidian University, China
[†]Department of ECE, Michigan State University, USA
Emails: {chaoyang[2], jiangqixdu[4], xfchen[5]}@xidian.edu.cn, {danpingli[1], chengzhouli[3]}@stu.xidian.edu.cn, jfma[6]@mail.xidian.edu.cn, renjian[7]@msu.edu.

***Abstract*—The replication and dissemination of multimedia data become increasingly convenient and efficient, so a lot of redundant multimedia data, especially image files, have been generated and stored on the Internet. Therefore, it is necessary to perform deduplication of images. However, the existing deduplication methods of regular files are hash-based, which cannot be applied to the deduplication of images. The deduplication of images faces following three challenges: it needs to check duplicates fuzzily; it needs to verify the ownership of similar images; it needs perceptual image quality assessment. Aiming at these challenges, we propose a scheme named the Client-based Security Provable Deduplication of Multimedia Data (CSPD). The proposed CSPD scheme is capable of responding to the above challenges. Furthermore, it meets provable security requirements. Our extensive simulation and performance analysis show that the CSPD can check duplicates accurately and assess the perceptual quality of distorted images. Moreover, the proposed CSPD is more efficient than the existing schemes in communication bandwidth and storage spaces.**

Keywords—proof of ownership; image deduplication; perceptual hash; discrete cosine transform

I. INTRODUCTION

In the Internet age, a large number of users store their data on the cloud server. In order to save the communication bandwidth and the storage capacity in this storage mode, researchers have proposed a lot of client-based deduplication methods [1-3]. It is reported that business applications can achieve deduplication ratios from 1:10 to as much as 1:500, which will result in disk and bandwidth savings of more than 90% [4].

However, the client-based deduplication raises some new security issues. Halevi *et al*. [5] recently discovered some new attacks to the client-based deduplication system, in which an attacker can obtain the entire file from the server simply by learning just a small piece of information about the file, namely its hash value. To solve this problem, Halevi *et al*. [5] firstly proposed a new cryptographic primitive PoW, the proof of ownership. In PoW, the client interacts with the server to prove that it owns a file which is completely the same as the file on the server. As far as we know, the proof of ownership is a general security issue in all client-based deduplication researches.

The replication and dissemination of multimedia data become increasingly convenient and efficient, so a lot of redundant multimedia data, especially image files, have been generated and stored on the Internet. Being different from regular files, images need more communication bandwidth and storage capacity during the process of transmission and storage. Therefore, how to perform the client-based deduplication of images efficiently and securely becomes an urgent problem.

Different from the existing client-based deduplication of regular files, the client-based deduplication of images faces many new challenges.

Challenge 1: the client-based deduplication of images needs to check duplicates fuzzily.

Duplicates of the original image include images exactly the same as the original one, and images transformed from the original one. A transformed image is not exactly the same as the original image, but is similar to the original image from the human perception [6,7]. There are a lot of transformed images on the cloud server. Therefore, it is necessary to judge the similarity between the original image and the transformed image to check duplicates fuzzily.

However, most of the existing client-based deduplication methods [1-3] use hash values of files to check duplicates accurately. These methods cannot respond to the new challenge of checking duplicates fuzzily.

Challenge 2: the client-based deduplication of images needs perceptual image quality assessment.

Except for duplicates check, the client-based deduplication of images also needs perceptual image quality assessment. In other words, in order to minimize losses of the image's quality, the server generally chooses one high-quality image to store in its database. If the server is requested to send a low-quality image to the client, it also can transform the high-quality image into a low-quality image [8]. Therefore, the client-based deduplication of images needs perceptual image quality assessment to choose suitable images, especially the high-quality image to store on the server.

However, perceptual image quality assessment was not taken into consideration in all of the recent researches about the client-based deduplication of images. Even in researches about the non-client-based deduplication of images, perceptual image quality assessment is rarely taken into consideration [9-12]. Although few methods of perceptual image quality assessment were mentioned in researches [11, 12], these methods cannot

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

meet the requirements of efficiency and accuracy. Moreover, the performance of these methods is poor in terms of correlation with human perception of quality. Therefore, there is no suitable method of perceptual image quality assessment in the field of the client-based deduplication of images.

Challenge 3: the client-based deduplication of images needs to verify the ownership of similar images.

The client-based deduplication of images needs duplicates check and perceptual image quality assessment. Furthermore, it needs to verify the ownership of similar images. In other words, when finding the client's image similar to one image stored on the server, the server will request the client to prove that it owns all the original data of that similar image. Therefore, in current network environment where all kinds of malicious attacks are constantly emerging [13,14], the client-based deduplication of images needs to verify the ownership of similar images to ensure that the attacker cannot steal the image from the server simply by learning just a small piece of information about the image.

However, as far as we know, none of the existing literatures takes the ownership verification of similar images into consideration. For example, Halevi *et al.* [5] firstly proposed the proof of ownership to prevent the attacker from obtaining the entire file simply by checking the hash value of the file. However, in the scenario of similar images deduplication, the image stored on the server is not exactly the same as the image stored on the client. Consequently, the method of Halevi *et al.* cannot respond to the new challenge of ownership verification of similar images.

Recently, few researches about the client-based deduplication of images have been done by researchers. The methods proposed by Gang *et al.* [15] and Rashid *et al.* [16] can only perform the deduplication of identical images. These methods are essentially the same as the client-based deduplication of regular files and cannot be applied to the new scenario. Another scheme proposed by Li *et al.* [17] is about the deduplication of similar images. However，it has these problems needed to be considered further: 1) the average phash [18] used to check duplicates fuzzily is less accurate, so the scheme could result in user losing of their images; 2) the scheme does not take perceptual image quality assessment into consideration, which could cause irreparable loss of images quality. For example, supposing that one low-quality image is stored on the server, when the client wants to upload a high-quality image, it will be informed to perform the deduplication. Therefore, when the client retrieves the image from the server, it can only get the low-quality one; 3) last but not least, the scheme does not carry out the ownership verification of similar images. The scheme will result in the attacker easily obtaining images from the server by the phash of images, which has serious security vulnerabilities [5].

None of above deduplication methods could respond to these new challenges faced by the client-based deduplication of images. Aiming at problems of above methods and these new challenges, we propose a scheme named the Client-based Security Provable Deduplication of Multimedia Data (CSPD). In the scheme, an algorithm named the Modified DCT-based Perceptual Image Hash (D-phash) is designed to improve the accuracy of the duplicate check. Also, a new protocol named Verify the Second Phash (VSP) is proposed to verify the ownership of similar images. Moreover, a method named the Perceptual Quality Assessment of Images (PQA) is designed for perceptual image quality assessment in the proposed scheme. The proposed CSPD scheme is capable of responding to the above challenges. Furthermore, it meets provably security requirements. Our extensive simulation and performance analysis show that the proposed CSPD can check duplicates accurately and assess the perceptual quality of distorted images. Moreover, the proposed CSPD is more efficient than the existing schemes in communication bandwidth and storage spaces needed to upload and store duplicated image.

II. CSPD FRAMEWORK

A. Basic Ideas

The proposed CSPD scheme consists of three parts: 1) Duplicate Check; 2) Proof of Ownership; 3) Quality Comparison. Fig. 1 presents the general framework.

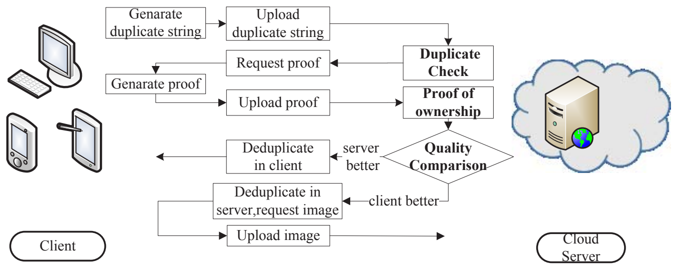

Fig. 1. General framwork of the proposed CSPD scheme.

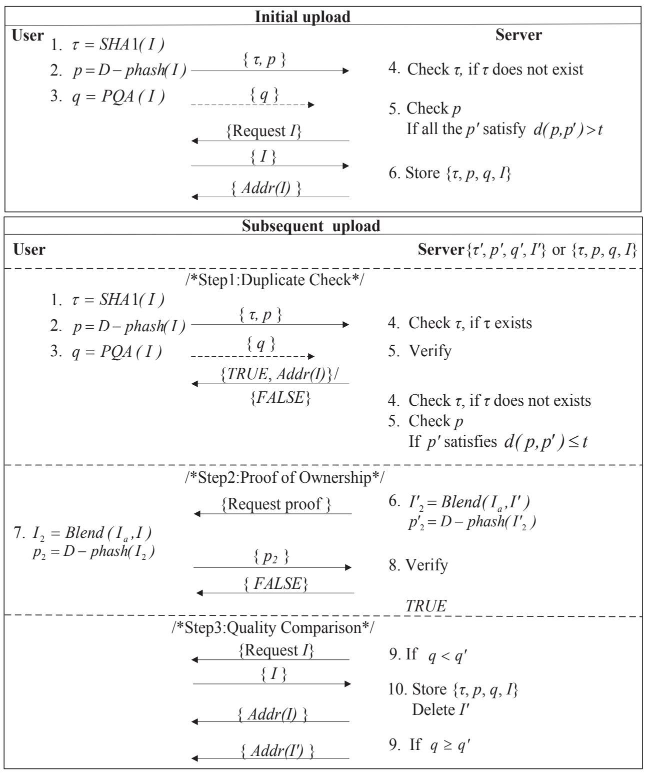

Fig. 2. The proposed CSPD scheme.

B. Detailed Construction

We are now ready to present the detailed construction of the proposed CSPD. The workflow is illustrated in Fig. 2.

1) Initial Upload

The workflow of the initial uploader is depicted in "Initial upload" part of Fig. 2.

The client side: the client calculates the hash value τ and the feature value p of the image I which it wants to upload to the server, $\tau = SHA1(I)$ and $p = D - phash(I)$. After uploading τ and p, the client calculates the quality $q = PQA(I)$ and uploads q to the server, where the quality q means the perceptual quality of the image I.

The server side: The server checks whether the hash value τ already exists in its database. If the server finds that τ does not exist in the database, it will calculate the Hamming Distance between the feature value of the image I and feature values of all the images stored on the server in sequence. If all the distances are greater than the threshold t, the server will request the client to upload the image I. The threshold is set as $t = 5$ according to performance analysis of the D-phash algorithm. After receiving the image I, the server stores the image I, the hash value τ, the feature value p and the quality q in the database. Then the server returns the address of the image I to the client.

2) Subsequent Upload

The workflow of the subsequent uploader is depicted in "Subsequent upload" part of Fig. 2.

• Step_1: Duplicate Check

The client side: The client calculates the hash value τ and the feature value p of the image I, and uploads τ and p to the server. The D-phash presented in Algorithm 1 is the key part of the "Duplicate Check".

Algorithm 1 Modified DCT-based Perceptual Image Hash($\mathrm{D-phash}$)

Input: The image I.
Output: The feature value p.

1: Read the image I:
$img\_bgr = imread(I)$.
2: Convert the image to grey scale:
$img\_gray = cvtColor(img\_bgr)$.
3: Resize the image:
$img\_dst = resize(img\_gray)$,
where the size of the img_dst is set as 64×64.
4: Compute the DCT matrix F:
$F = dct(img\_dst)$,
where the size of F is 64×64.
5: Select low-frequency DCT matrix F^l:
Keep the top-left 8×8 of F, which means, $F^l = F(0,...,7;0,...,7)$.
6: Compute the mean value of F^l:
$mean = (\sum F^l(i,j) - F^l(0,0))/63$,
where $0 \le i \le 7$, $0 \le j \le 7$, $F^l(0,0)$ is the DC coefficient.
7: Normalize F^l into a binary form:

$$p(i,j) = \begin{cases} 0, & F^l(i,j) \le mean \\ 1, & F^l(i,j) > mean \end{cases}$$

8: Construct the feature value: $p = p(i,j)$.
9: **return** p.

The server side: The server checks whether the hash value τ already exists in its database. If the server finds that τ is the same as an existing hash value stored in the database, the server and the client will carry out the verification protocol PoW [5]. If the client passes the verification, it does not have to upload the image I to the server; otherwise, the client is rejected. If the server finds that τ does not exist in the database, it will calculate the Hamming Distance between the feature value of the image I and feature values of all the images stored on the server in sequence. If one distance is less than the threshold t, the server and the client will carry out the VSP protocol.

• Step_2: Proof of Ownership

The VSP protocol presented in Algorithm 2 is used to verify the ownership of similar images.

Algorithm 2 Verify the Second Phash(VSP)

Challenge:
1: The server S randomly selects a auxiliary image I_a, it sends id of I_a to the client and requests the client C to provide the proof.
2: The server S reads the auxiliary image I_a and the image I', where I' is the image stored on the server which is similar to the image I.
3: The server S resizes the auxiliary image I_a:
$size(width\_I_a, hight\_I_a) = size(width\_I', hight\_I')$.
4: Let the blending parameter α=0.5, the server S genarates the blended image $I'_2 = \mathrm{Blend}(I', I_a, \alpha)$, which means $I'_2 = 0.5 \times I' + 0.5 \times I_a$.
5: The server S computes the feature value $p'_2 = \mathrm{D-phash}(I'_2)$.

Response:
1: After receiving id of the auxiliary image I_a, the client C reads the auxiliary image I_a corresponding to the id, and reads the image I .
2: The client C resizes the size of I_a:
$size(width\_I_a, hight\_I_a) = size(width\_I, hight\_I)$.
3: Let $\alpha = 0.5$, the client C genarates the blended image $I_2 = \mathrm{Blend}(I, I_a, \alpha)$, which means $I_2 = 0.5 \times I + 0.5 \times I_a$.
4: The client C computes the feature value $p_2 = \mathrm{D-phash}(I_2)$, then it sends p_2 to the server.

ProofCheck:
1: After receiving the feature value p_2, the server computes the Hamming Distance $d = \mathrm{HammingDistance}(p_2, p'_2)$.
2: **if** $d \le t$ **then**
3: output True
4: **else**
5: output False
6: **end if**

Fig. 3. The image blending process

The image blending process in the VSP protocol is depicted in Fig. 3. Also, the feature values p and p_2 generated by the D-phash algorithm are presented in Fig. 3. Obviously, there is no relation between p and p_2, which means, p_2 cannot be derived from p. This conclusion will be demonstrated in section III.

• Step_3: Quality Comparison

The server judges whether q is less than q', where q is quality of the image I which the client wants to upload and q' is quality of the image I' stored on the server. If $q < q'$, the server will request the client to upload the image I. After receiving the image I, the server deletes the image I' previously stored on the server. Then the server stores the image I in its database and returns the address of the image I to the client. If $q \geq q'$, the server will return the address of the image I' to the client. The client calculates quality q after uploading τ and p, and $q = PQA(I)$. The PQA is designed based on the BRISQUE [19], it uses scene statistics of locally normalized luminance coefficients to quantify possible losses of "naturalness" in the image due to the presence of distortions.

There are four cases in the proposed CSPD scheme. The initial uploader will encounter the situation where none of the images stored on the server is similar to the image I which the client wants to upload (no similar). In this case, the client needs to upload the image to the server. The subsequent uploader will encounter the following three situations: one image stored on the server is the same as the image I (same); one image stored on the server is similar to the image I and its quality is better than that of the image I (server better); one image stored on the server is similar to the image I and its quality is worse than that of the image I (client better). In the first two cases, the client does not need to upload the image I. Only in the last case, the client needs to upload the image I. Therefore, the proposed CSPD scheme is capable of storing high-quality images on the server, and it saves a lot of network bandwidth and storage spaces.

III. SECURITY ANALYSIS

In this section, we will analyze the security of ownership verification of the proposed CSPD scheme. In the CSPD, the server requests the client to upload p_2 to prove its ownership of the image I, where p_2 is the feature value of the blended image I_2. The first bit of p_2 is 1, and the threshold is $t = 5$.

Theorem 1. In the proposed CSPD scheme, Alice can pass the ownership verification with probability $P_{pass} \leq P_G$ if she only had the feature value p of the image I, where P_G is the probability that Alice can obtain 58 bits (except the first bit) of p_2, which means $P_{pass} \leq 1/2^{58}$.

Proof: Alice needs to get p_2, and p and p_2 must satisfy $d(p, p_2) < t$. Alice has two methods: if there is a relation between p and p_2, Alice will derive p_2 from p; otherwise, Alice will launch a brute force attack to obtain 58 bits of p_2.

If $F^l(i,j)$ and $F^l{}_2(i,j)$ satisfy $F^l{}_2(i,j) > F^l(i,j)$, and $p(i,j) = 1$, then $F^l{}_2(i,j) > mean$ of $F^l{}_2$, which means $p_2(i,j) = 1$; if $F^l(i,j)$ and $F^l{}_2(i,j)$ satisfy $F^l{}_2(i,j) \leq F^l(i,j)$, and $p(i,j) = 0$, then $F^l{}_2(i,j) \leq mean$ of $F^l{}_2$, which means $p_2(i,j) = 0$. Therefore, in these two cases, if Alice knows the relation between $F^l{}_2(i,j)$ and $F^l(i,j)$, namely she knows whether $F^l{}_2(i,j) > F^l(i,j)$ or not, then she can derive $p_2(i,j)$ from $p(i,j)$.

The server selects the auxiliary image I_a randomly, so there is no relation between $I(i,j)$ and $I_a(i,j)$, where $I(i,j)$ is any pixel of the image I and $I_a(i,j)$ is any pixel of the image I_a. $I_2(i,j) = 0.5 \times I(i,j) + 0.5 \times I_a(i,j)$, so the relation between $I_2(i,j)$ and $I(i,j)$ depends on the relation between $I(i,j)$ and $I_a(i,j)$. Therefore, there is no relation between $I(i,j)$ and $I_2(i,j)$, namely it is impossible to know whether $I(i,j) \geq I_2(i,j)$ or not.

It is known that $F = CIC^T$, $F_2 = CI_2C^T$, where

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & & & \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix}, I = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & & & \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}, I_2 = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \vdots & & & \\ y_{n1} & y_{n2} & \dots & y_{nn} \end{bmatrix},$$

$$CI(i,j) = c_{i1}x_{1j} + c_{i2}x_{2j} + \dots + c_{in}x_{nj},$$

$$CI_2(i,j) = c_{i1}y_{1j} + c_{i2}y_{2j} + \dots + c_{in}y_{nj}.$$

Therefore, there is no relation between $CI(i,j)$ and $CI_2(i,j)$, namely it is impossible to know whether $CI(i,j) \geq CI_2(i,j)$ or not. In a similar way, there is no relation between $CI(i,j)C^T$ and $CI_2(i,j)C^T$. Therefore, there is no relation between $F^l(i,j)$ and $F^l{}_2(i,j)$, namely it is impossible to know whether $F^l{}_2(i,j) > F^l(i,j)$ or not.

Therefore, Alice cannot derive $p_2(i,j)$ from $p(i,j)$, which means $P_{pass} \leq 1/2^{58}$. It is believed that the probability is small enough to be ignored, so Alice cannot pass the ownership verification.

IV. PERFORMANCE EVALUATION

In this section, we will compare the proposed CSPD scheme with the existing method SPSD [17].

TABLE I. CLIENT SIDE CONFIGURATION

Item	Parameter
DISK	40GB
CPU	SINGLE-CORE
MEMORY	1GB
OS	UBUNTU 15.10

TABLE II. SERVER SIDE CONFIGURATION

Item	Parameter	Item	Parameter
SERVER	ALI CLOUD	CPU	SINGLE-CORE
LOCATION	QINGDAO	MEMORY	4GB
Server ID	i-281kfyxga	OS	UBUNTU 14.04
DISK	40GB	BANDWIDTH	5MBPS

A. Testing Environment and Scheme

We build the client locally and configure the parameters according to Table I. We deploy the server in Qingdao, and the distance between Qingdao and Xi'an, the client's location, is 1058.6km. The cloud server parameters are configured according to Table II. We use the OpenCV2.4.12 library for image processing and use C++ for the implementation.

We use images in the LIVE database [20]. LIVE was established by LIVE (Laboratory for Image and Video Engineering), and it is usually used for image quality assessment. Four types of distortions in the LIVE are considered: Fast Fading, Gaussian Blur, JPEG2000 Compression, and JPEG Compression.

B. Simulation Results

Characteristics comparisons between the CSPD and the SPSD are depicted in Table III.

TABLE III. COMPARISONS OF CHARACTERISTICS BETWEEN THE CSPD AND THE SPSD

Scheme	**Phase**		
	Duplicate Check	***Proof of Ownership***	***Quality Comparison***
SPSD	YES	NO	NO
CSPD	YES	YES	YES

TABLE IV. COMPARISONS OF CHARACTERISTICS BETWEEN THE D-PHASH USED IN THE CSPD AND THE A-PHASH USED IN THE SPSD

Scheme	**Time/s**	**Accuracy**
SPSD (A-phash)	0.0005	low
CSPD (D-phash)	0.0011	high

- **Duplicate Check**

The perceptual hash algorithm is used to judge the similarity between the image which the client wants to upload to the server and the images stored on the server. The CSPD uses the D-phash algorithm and the SPSD uses the A-phash (average phash) algorithm. Characteristics comparisons between the D-phash and the A-phash are depicted in Table IV, in which "Time/s" is the mean of 20 trials and "Accuracy" includes two aspects of the robustness and the discriminative capability. Because the amount of all the experiment data is huge and JPEG Compression is one of the most commonly used distortion types, we only present specific experiment data of JPEG Compression.

Duplicate Check Time. This is the time to execute duplicate check of the SPSD or the CSPD. We select 20 sets of JPEG Compression images to give the SPSD and the CSPD 20 trials respectively. Results are depicted in Fig. 4. The time difference between the SPSD and the CSPD in duplicate check phase is only about 1ms, which is very small.

Robustness. When judging the similarity between two similar images, a perfect perceptual should generate a distance which is less than the threshold. We select all 29 sets of JPEG Compression images. There are 7~9 similar images in each set. Robustness of the D-phash and the A-phash are depicted and summarized in Fig. 5. The D-phash is more robust than the A-phash. Consequently, the deduplication rate of the CSPD is higher than that of the SPSD.

Discriminative Capability. When judging the similarity between two different images, a perfect perceptual should generate a distance which is greater than the threshold. We select all the 29 reference images. Discriminative capabilities of the D-phash and the A-phash are depicted and summarized in in Fig. 6. Discriminative capability of the D-phash is better than that of the A-phash. Therefore, the CSPD is more reliable than the SPSD.

- **Proof of Ownership**

Ownership Verification Time. This is the time to execute Algorithm 2. We select 20 sets of images in each types of distortion. Times to execute Algorithm 2 are depicted in Fig. 7. The time range is between 40ms and 60ms, which is very small compared with the time of the whole scheme. However, the SPSD does not carry out the ownership verification.

Overall, compared with the SPSD, the CSPD can check duplicates more accurately. Furthermore, it meets provable security requirements without increasing excessive computation and time.

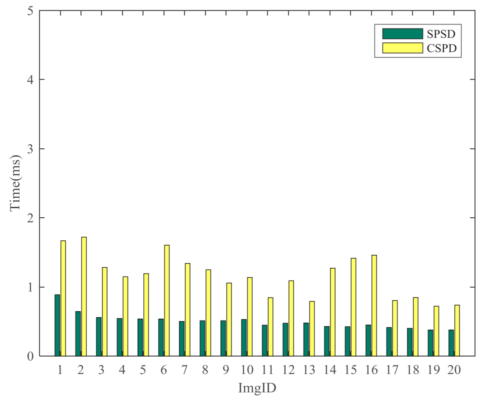

Fig. 4. Comparison of duplicate check time between the proposed CSPD scheme and the existing SPSD scheme.

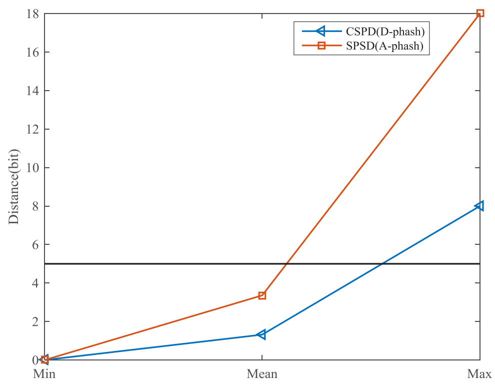

Fig. 5. Comparison of robustness between the D-phash and the A-phash.

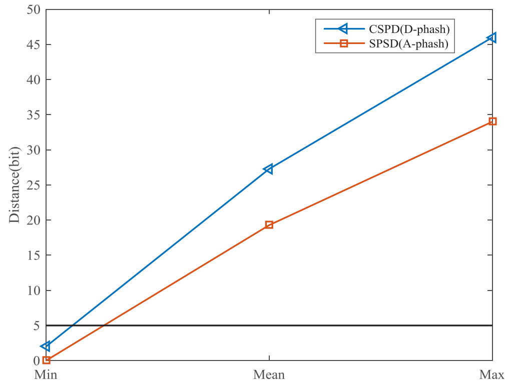

Fig. 6. Comparison of discriminative capability between the D-phash and the A-phash.

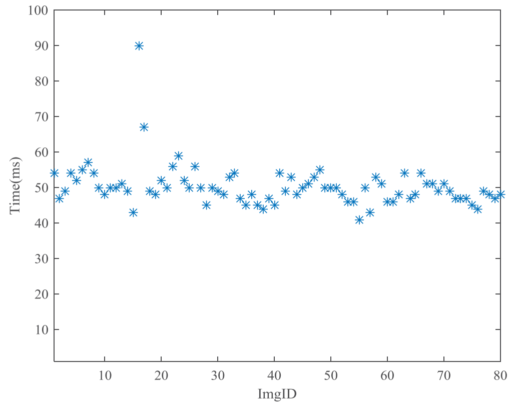

Fig. 7. Ownership verification time of the proposed CSPD scheme.

V. Conclusion

Aiming at the new challenges faced by the client-based deduplication of images, we propose a scheme named the Client-based Security Provable Deduplication of Multimedia Data (CSPD). We provide rigorous security analysis and extensive performance evaluation to show that the proposed CSPD scheme meets provable security requirements and it can check duplicates accurately and store the image which has the best perceptual image quality on the server. Furthermore, The CSPD saves a lot of bandwidth and spaces needed to upload and store duplicated data. Moreover, construction of the image deduplication method protecting confidentiality of images is in our future work.

References

[1] Pietro R D, Sorniotti A. Boosting efficiency and security in proof of ownership for deduplication//Proceedings of the 7th ACM Symposium on Information, Compuer and Communications Security. Seoul, Korea, 2012:81–82.

[2] Zheng Q, Xu S. Secure and efficient proof of storage with deduplication//Proceedings of the 2th ACM Conference on Data and Application Security and Privacy. New York, USA, 2012:1–12.

[3] Yang C, Ren J, Ma J. Provable ownership of files in deduplication cloud storage//Proceedings of the IEEE Global Communications Conference. Atlanta,GA, USA, 2013:2457-2468.

[4] M. Dutch and L. Freeman, “Understanding data de-duplication ratios,” http://www.snia.org/, 2009.

[5] Halevi S, Harnik D, Pinkas B, et al. Proofs of ownership in remote storage systems//Proceedings of the 18th ACM Conference on Computer and Communications Security. Chicago, Illinois, USA, 2011:491-500.

[6] Joly A, Buisson O, Frelicot C. Content-based copy retrieval using distortion-based probabilistic similarity search. IEEE Transactions on Multimedia, 2007, 9(2):293-306.

[7] Ming C, Wang S, Yun X, et al. FAIDA: a fast and accurate image deduplication approach. Journal of Computer Research and Development, 2013, 50(1):101-110.

[8] Katiyar A, Weissman J. ViDeDup: an application-aware framework for video de-duplication//Proceedings of the 3th USENIX Conference on Hot topics in storage and file systems. Portland, Oregon, USA, 2011: 31-35.

[9] Yang X, Su G, Chen J, et al. Large scale identity deduplication using face recognition based on facial feature points//Proceedings of the 6th Chinese Conference on Biometric Recognition. Beijing, China, 2011:25-32.

[10] Xu J, Zhang W, Ye S, et al. A lightweight virtual machine image deduplication backup approach in cloud environment//Proceedings of the 38th IEEE Signature Conference on Computers, Software and Applications. Vasteras, Sweden, 2014:503-508.

[11] Ramaiah N P, Mohan C K. De-duplication of photograph images using histogram refinement//Proceedings of the IEEE International Conference on Recent Advances in Intelligent Computational Systems Recent Advances in Intelligent Computational Systems. Trivandrum, India, 2011:391-395.

[12] Chen M, Wang Y, Zou X, et al. A duplicate image deduplication approach via Haar wavelet technology//Proceedings of the 2th IEEE International Conference on Cloud Computing and Intelligent Systems. Hangzhou, China, 2012:624-628.

[13] Xiao L, Xie C, Chen T, et al. A Mobile Offloading Game Against Smart Attacks. Access IEEE, 2016, 4:2281-2291.

[14] L. Xiao, D. Xu, C. Xie, N. B. Mandayam, and H. V. Poor. Cloud Storage Defense Against Advanced Persistent Threats: A Prospect Theoretic Study. IEEE Journal on Selected Areas in Communications, accepted.

[15] Gang H, Yan H, Xu L. Secure image deduplication in cloud storage. Information and Communication Technology. Springer International Publishing, 2015, 9357:243-251.

[16] Rashid F, Miri A, Woungang I. Secure image deduplication through image compression. Journal of Information Security and Applications, 2016, s 27–28:54-64.

[17] Li X, Li J, Huang F. A secure cloud storage system supporting privacy-preserving fuzzy deduplication. Soft Computing, 2015, 20(4):1437-1448.

[18] Zauner C. Implementation and Benchmarking of Perceptual Image Hash Functions. Revista Musical Chilena, 2011, 65(215):71-72.

[19] Mittal A, Moorthy A K, Bovik A C. No-reference image quality assessment in the spatial domain. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2012, 21(12):4695-4708.

[20] H.R. Sheikh, Z.Wang, L. Cormack and A.C. Bovik, “LIVE Image Quality Assessment Database Release 2” http://live.ece.utexas.edu/research/quality.