A Cloud-User Watermarking Protocol Protecting the Right to Be Forgotten for the Outsourced Plain Images

Xiaojuan Dong, CAS Key Laboratory of Electro-magnetic Space Information, Chinese Academy of Sciences, University of Science and Technology of China, Hefei, China

Weiming Zhang*, CAS Key Laboratory of Electro-magnetic Space Information, Chinese Academy of Sciences, University of Science and Technology of China, Hefei, China

Xianjun Hu, CAS Key Laboratory of Electro-magnetic Space Information, Chinese Academy of Sciences, University of Science and Technology of China, Hefei, China

Keyang Liu, CAS Key Laboratory of Electro-magnetic Space Information, Chinese Academy of Sciences, University of Science and Technology of China, Hefei, China

ABSTRACT

Cloud storage dramatically benefits people from freeing their local storage space, while bringing the separation of the data ownership and private manipulation. Hence, it is difficult for the cloud user to make sure that the cloud storage provider (CSP) has obeyed the request of deletion to remove all corresponding data. To solve the issue technically, in this paper, we propose an interactive cloud-user watermarking protocol (CUW) based on the homomorphic encryption. To meet security requirements, the encrypted watermark is embedded into encrypted data. Moreover, to enjoy the convenient cloud services, the uploaded data are eventually stored in the cloud server in the form of plain text. The performance of the CUW protocol is evaluated through a prototype implementation.

KEYWORDS

Right to Be Forgotten, Watermarking, Crime Forensics, Secure Deletion, Cloud Storage

1. INTRODUCTION

With the rapid growth of data scale, the demand for storage space is likewise increasingly growing. In this trend, cloud storage service has been recently presented as a service. This kind of service provides people with a lot of cheap and unlimited storage space. For example, Amazon Web Services and Google Cloud Storage offer cloud storage solutions to customers around the world, reducing the need of local devices' storage space. Despite the tremendous benefits, the cloud user's data, held in remote cloud storage, are absolutely beyond the user's

* Corresponding author.

E-mail address: xjuadong@mail.ustc.edu.cn (X. Dong), zhangwm@ustc.edu.cn (W. Zhang), hxj2012@mail.ustc.edu.cn (X. Hu), lky58587@mail.ustc.edu.cn (K. Liu).

control. It is necessary to guarantee the assured deletion (Ramokapane, Rashid, & Such, 2016) for cloud users. The undeleted data may unexpectedly appear later, and thus exposes the user's private information. The challenge of realizing assured deletion is that we have to trust in the CSP, who will completely delete data according to contract. It is a typical and practical trend among CSPs to store multiple backups of data over different online or offline servers for fault tolerance. One specific case is a fact that after receiving the request of deletion, CSPs may not actually remove all backup copies even though they have deleted the data in the current cloud server. Therefore, it is difficult to confirm that data have been forgotten by CSPs. How to completely remove data and maintain cloud users' right to be forgotten has become an urgent problem.

Great importance has been attached to the right to be forgotten by many organizations. They introduced a series of security policies and laws. As early as 1995, the EU passed the 1995 Data Protection Directive, under which the data controller is required to remove the personal data of an individual upon request (Europea, 1995). This is the genesis of the right to be forgotten, which means that any organization is obligated to remove a customer's personal data upon request. On May 13, 2014, the European Court of Justice compelled Google to remove links to a 1998 newspaper article about a Spanish man's bankruptcy (Kropf, 2014), upholding the right to be forgotten on the Internet. Cybersecurity Law of the People's Republic of China has come into effect since June 1, 2017, which states that: Network operators collecting and using personal information shall abide by the principles of legality, propriety and necessity and obtain the consent of the person whose data is gathered. In reality, these rules are included in the service contracts, cloud users still have to trust heavily in CSPs without any technical guarantee.

For assured deletion, a typical prior work in this area focuses on encryption technology (Perlman, 2005; Tang, Lee, Lui, & Perlman, 2012; Priebe, Muthukumaran, O'Keeffe, Eyers, Shand, Kapitza, & Pietzuch, 2014). According to (Tang, Lee, Lui, & Perlman, 2012), assured deletion makes the outsourced data permanently inaccessible to anyone upon the request of data deletion. In (Tang, Lee, Lui, & Perlman, 2012), a data owner first encrypts data and then sends the encrypted data to cloud servers managed by a CSP. Relevant encryption keys are preserved by the owner or managed by a third party. Finally, the deletion operation is artfully achieved by destroying encryption keys. As a result, the data can no longer be decrypted, not to mention being accessible. This kind of data encryption scheme gives cloud users some control over the removal of their data.

However, ciphertext computing services are very complex and impractical. By outsourcing plain data to cloud servers, users can fully enjoy fast cloud computing services, such as using an image to search other similar images, editing images in large-scale image libraries and so on. Therefore, we advocate uploading plain data to cloud servers. Unfortunately, there is no specific assured deletion scheme of plain data stored in cloud.

We observe that it is difficult to prove that the assured deletion of data has been done, but it is rather easy to prove that the deletion has not been done once you encounter your data that should have been deleted. For example, a cloud user's data is marked with a CSP's unique identifier, and then stored in the CSP's server. Once the cloud user requires to delete his/her data and meets the data later, he/she can claim that the CSP does not perform the complete deletion according to service contracts, and he/she can use the identifier of CSP to prove CSP's crime. Throughout our consideration, assured deletion of plain cloud data succeeds when all copies of the data disappear forever. In turn, assured deletion fails when a copy of the data appears and the dishonest CSP can

be traced with the identifier. Motivated by this, this paper surveys and exploits the traceability property of digital watermarking for identifying dishonest CSPs.

Watermarking techniques have been widely investigated in copy deterrence and tracing down the distribution of illegal replicas in (Jin, 2009; Xia, Wang, Zhang, Qin, Sun, & Ren, 2016). In buyer-seller scenarios (Memon, & Wong, 2001; Kuribayashi, & Tanaka, 2005; Frattolillo, 2017), the seller embeds the buyer's identity as a watermark into the content before it being sold to the buyer so that this seller can retrieve this buyer's identity when he/she encounters a redistributed copy. The retrieved buyer's identifier can be employed as the sufficient evidence of the buyer's illegal distribution of behavior. The cloud-user scenario is similar to the buyer-seller scenario. If the buyer-seller protocol is directly adopted in this paper, the cloud user will be regarded as the seller while the CSP will be as the buyer. But the retrieved CSP's identifier is insufficient for our needs. In the cloud-user scenario, after a cloud user uploads data to a cloud server, the uploaded data has been marked with the corresponding CSP's identifier. This cloud user can touch the same data as the one stored in the cloud server by downloading his/her data. As the downloaded data contains the CSP's identifier, the cloud user can intentionally distribute the copy, and later accuses the innocent CSP for benefits. To hinder the false accusation, we wan buyer-seller protocol adopted again, but in reverse.

In this paper, we present a cloud-user watermarking (CUW) protocol that supports outsourcing plain data while technically, upholding the user's right to be forgotten. Users' data are lodged in cloud servers in plain-text forms. Before uploaded, plain data are invisibly marked with the corresponding CSP's identifier, a unique identity watermark. Then, the plain data is bound with the user's unique identity watermark before downloaded. In order to ensure fairness to both parties, the generation and embedding operations of watermark are accomplished by a watermark certification authority (WCA). Both the CSP and the user do not know their own watermarks. Meanwhile, both of them do not have any idea about each other's watermark. They are unable to recreate the data copy containing the other's watermark. Once an illegal distributed copy appears, the extracted watermark can reveal the untruthful distributor without being framed. Although WCA is the trusted party, plain data is unobservable to WCA for it may be attacked maliciously. In our protocol, all clear data are encrypted during all transmissions for protecting privacy. Lastly, a receiver decrypts ciphertext data and obtains plain data with his/her identity watermark embedded. Our proposed protocol employs off-the-shelf cryptographic schemes involved with homomorphic encryption (Rivest, Adleman, & Dertouzos, 1978).

Our Contributions. The main contributions are summarized as follows:

- To the best of our knowledge, the proposed protocol is the first framework that achieves the assured deletion of plain data and protect users' right to be forgotten.
- This paper can provide technical evidence to accuse the illegal party. Both the cloud user and the CSP are treated with fairness and justice.
- We implement a working prototype of the CUW protocol over images and conduct experiments to evaluate its performance.

2. PROBLEM FORMULATION

2.1. Protocol Model

In this article we are particularly interested in images, a popular kind of multimedia data, and the

cloud user is specified as the image owner who can enjoy cloud storage service. The protocol model in this paper involves three different roles: the image owner, the CSP and the trusted WCA.

Image owner holds a large volume of images that are usually of extremely large size. The image owner outsources his/her images to cloud and wants to manipulate the images, like downloading them or deleting them. In our scheme, only the image owner can retrieve the images. In addition, multiple users are allowed to take back the owner's images for sharing. But before that, the image owner needs to send authentication information to cloud. The CSP will verify the authentication of the user, who requests downloading the owner's image.

CSP offers and manages cloud storage servers, which have a huge amount of storage space. A cloud server at least has the thin-cloud interface (Vrable, Savage, & Voelker, 2009), which allows the most basic data operations like access, storage and deletion.

WCA is a trusted agency who is responsible to generate and embed unique identifier watermarks for image owners and CSPs. Meanwhile, WCA will execute the arbitration through watermark extraction and adjudicate lawsuits against infringement. Note that WCA runs on another cloud computing server which is managed by other CSPs.

2.2. Threat Model

In the proposed scheme, anyone among the image owner, the CSP and the WCA can launch security problems. In this paper, three types of secure assumptions are particularly considered.

1) We assume that the CSP will correctly follow the protocol specification, but may not entirely delete all image copies at the owner's request. Unfortunately, those copies flow out from the CSP's servers.

2) We consider the faithless owner who obeys the protocol specification, but may frame the innocent CSP for interests. When the image owner is not trustworthy, two scenarios may occur as follows:

- The dishonest owner outsources his/her image to a CSP's cloud server, deliberately reserves copies of the image and distributes them. Later he/she claims that the CSP leaks his/her personal image without his/her permission. This problem will be settled through adopting the buyer-seller protocol.
- Adopted though the buyer-seller protocol is, the image owner can download his/her image and obtain the same image as the one lodged in the cloud server. The downloaded image has been embedded in the CSP's unique watermark during the uploading process. The owner may frame the CSP by disseminating his/her downloaded image. This kind of behavior should be further prevented. Thus, the buyer-seller protocol is adopted again, but in reverse.

3) Lastly, we rest on the assumption that WCA is selected by the image owner and CSP. WCA should be trusty so that its assertion can be used as evidence.

2.2. Design Goals

Our ultimate goal is to design an image protocol between image owners and CSPs. The proposed protocol indirectly supports assured deletion for plain images while maintaining owners' right to be forgotten. Concrete goals are formally declared as follows:

1) **Data Privacy**. As claimed in our protocol model, WCA is responsible for embedding watermarks in images. Though WCA is trusted, it may be viciously attacked. Hence, WCA has

no access to the image content like pixel values. Then, WCA is carefully designed so that identity watermarks can be embedded in encrypted images.

2) **Traceability**. Any copy of an image must be identifiable and traceable back to the distributor.

3) **Non-frameability**. The proposed watermark-based protocol is fair to both the owner and the CSP. Nobody can frame an honest party.

4) **The Right to Be Forgotten**. Absolute data deletion means that the deleted data permanently disappear. If an image owner encounters his/her deleted image, he/she can extract the watermark corresponding to a CSP's identity. Lastly, the owner can reasonably sue the CSP for his/her right to be forgotten.

3. PRELIMINARIES

In this section, we survey the state-of-the-art preliminaries of cryptography and watermarking which we will use in the construction of a practical system under our framework.

3.1. Homomorphism

The homomorphic property of public-key cryptosystems supports that an operation on a ciphertext space results in a meaningful operation on the plain message space without revealing the plain value. Given two plaintexts m_1 and m_2, an encryption scheme is said to be homomorphic if the encryption function *E* satisfies

$$\forall m_1, m_2 \in M: E[m_1 \odot_M m_2] = E[m_1] \odot_C E[m_2], \quad (1)$$

where *E*[] denotes the encryption operator for some operators $\odot_M$ in the plain domain *M* and $\odot_C$ in the encrypted domain *C*.

The well-known RSA cryptosystem (Rivest, Shamir, & Adleman, 1978) is a privacy homomorphism with respect to multiplication (Stinson, 2005). The Paillier cryptosystem (Paillier, 1999), and Paillier's generalization by Damgård-Jurik (Damgård, & Jurik, 2001) are additive privacy homomorphism, and map an addition in the plaintext domain to a multiplication in the ciphertext domain. Thus, this cryptosystem provides a way of applying any linear operator in the encrypted domain. For convenience sake, the Paillier cryptosystem is employed as the image encryption method in this paper. In order to make the homomorphic properties clear, we briefly introduce the definition of the Paillier cryptosystem.

Compute $N = p \times q$, where p and q are two large randomly generated prime numbers. Z_{N^2} denotes the set of integer numbers modulo N^2. $Z_{N^2}^*$ contains all the integers in Z_{N^2} which are primes of N^2. *m* is the plaintext from Z_N and thus, the corresponding ciphertext *c* will be mapped in $Z_{N^2}^*$. r is a randomly chosen integer from Z_N^* and g is a randomly chosen integer from $Z_{N^2}^*$. Use (*N*, *g*) as the public key and compute the encryption as equation (2).

$$c = E[m, r] = g^m r^N \ mod \ N^2 \quad (2)$$

According to equation (2), we can deduce several homomorphic properties as equation (3) and equation (4).

$$D\big[E[m_1, r_1], E[m_2, r_2]\big] = m_1 + m_2 \ \ mod \ N, \quad (3)$$

$$D[E[m_1, r_1]^k] = k m_1 \ mod \ N, \quad (4)$$

where *D*[] denotes the decryption operator.

When encrypting an image for security, we encrypt the image pixel by pixel using equation (2). Our watermarking scheme is based on this homomorphic property.

3.2. Dither Modulation

Quantization Index Modulation (QIM) is a classical watermarking algorithm (Chen, & Wornell, 2001). An information bit *w* is embedded by adjusting the host coefficient *x* in some transform domain to a scalar quantizer of step size Δ. The Dither Modulation (DM) is a kind of implementation for QIM. In the basic QIM scheme, before embedding *w*, a dither *d* is added to *x*, and then subtract d after embedding as

$$y = \begin{cases} Q_{\Delta-even}(x+d) - d, & if\ w = 0, \\ Q_{\Delta-odd}(x+d) - d, & if\ w = 1, \end{cases} \tag{5}$$

where $Q_{\Delta-even}(\cdot)$ represents the nearest even multiples of Δ to $\cdot$, $Q_{\Delta-odd}(\cdot)$ represents the nearest odd multiples of Δ to $\cdot$, and *d* is the dither value with the uniform distribution on $[-\Delta, \Delta]$.

4. PROPOSED PROTOCOL

In the proposed protocol, the public-key infrastructure (PKI) is available, such that each party has a public and private key pair certificated by the certification authority. We assume that the encryption function used in PKI is homomorphic with respect to the watermark insertion operation. In this section, we first define the roles and notations to be used throughout the rest part. Notations involved are listed in Table 1.

Table 1. Notations and Abbreviations

X	The original image
Y	The copy image of X
W_P	The vector of identity watermark belonging to the party P
O_W	The vector of ownership watermark belonging to the owner O
X_{W_P}	The watermarked image with W_P
X_{O_W}	The watermarked image with O_W
$X \oplus W$	Operator $\oplus$ represents the insertion of watermark W in the image X
ID_P	The identity of the party P
(pk_p, sk_p)	A pair of public key and secret key belonging to the party P
$Sign_P(\cdot)$	The signature signed with the party P's private key
E_{pk_P}	Encryption operation with the party P's public key
D_{sk_P}	Decryption operation with the party P's private key
$Table_{ID_O}$	The table recording the upload information for the owner of ID_O
A→B	A sends messages to B
n, n'	Upload request number n and download request number n'
$Detion_n$	The deletion signal related to the upload request number n

4.1. Overview of the Proposed Protocol

Given an image X, three roles, an image owner, a CSP's cloud server and WCA, all participate in the process of outsourcing X, as shown in Fig. 1(a).

The image owner first embeds his/her ownership watermark O_W into X and gets the watermarked X_{O_W}. O_W indicates that he/she is the true owner of X. After that, the owner computes the encrypted image $E_{pk_C}(X_{O_W})$ using the corresponding CSP's public key pk_C, and then sends $E_{pk_C}(X_{O_W})$ to WCA. After receiving $E_{pk_C}(X_{O_W})$, WCA generates a unique identity watermark W_C associated with the corresponding CSP and encrypts W_C with pk_C simultaneously. Next, WCA embeds $E_{pk_C}(W_C)$ into $E_{pk_C}(X_{O_W})$ and obtains the encrypted and watermarked image $E_{pk_C}(X_{W_C})$. Finally, $E_{pk_C}(X_{W_C})$ is delivered to the cloud server by WCA. After receiving $E_{pk_C}(X_{W_C})$, the cloud server decrypts it with the corresponding CSP's private key sk_C and then procures the plain image X_{W_C}.

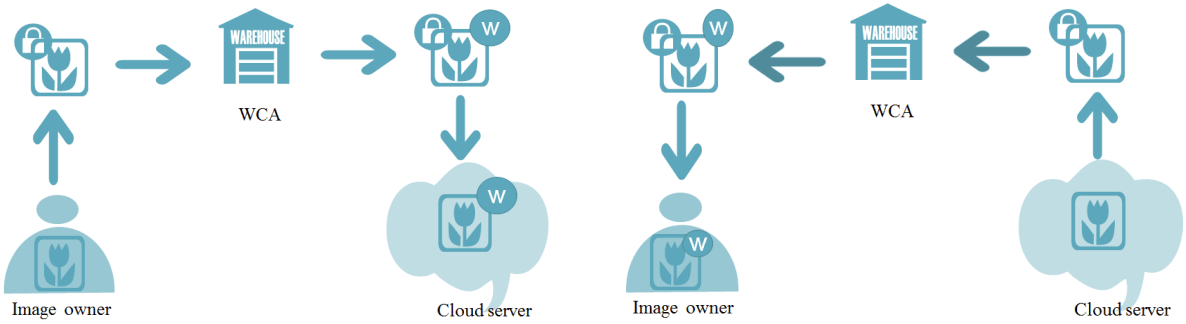


Fig. 1. Framework of the proposed protocol. (a) The image outsourcing process; (b) The image retrieval process

In the image outsourcing process, the image owner is unable to recreate the same outsourced image as the one stored in the cloud server. The owner can require CSP to remove all copies of his/her image X at any time. Once an illegal image Y, a copy of his/her deleted image X, appears out of expectation, the owner can identify the specific CSP from the watermark extraction. The extracted watermark of ownership O_W from Y can affirms Y belongs to him/her. The extracted identity watermark W_C from Y means that Y originates from the cloud server operated by the dishonest CSP. Furthermore, the owner can give the evidence to WCA.

After the owner retrieves his/her image X_{W_C} from the cloud server, he/she may disseminate X_{W_C} and deliberately accuse the CSP of dishonesty and betrayal. Hence, we specifically design the image retrieval process. The interactive activities among the owner, the cloud server and WCA are illustrated in Fig. 1(b).

When the cloud server receives a request from the image owner to download the image X_{W_C}, the cloud server uses the owner's public key pk_O to encrypt $X_{W_C}$and transmits $E_{pk_O}(X_{W_C})$ to WCA. Then WCA generates a unique identity watermark W_O of the owner and encrypts it with pk_O. Next, the encrypted watermark $E_{pk_O}(W_O)$ is embedded into $E_{pk_O}(X_{W_C})$, and then the encrypted and watermarked image $E_{pk_O}(X_{W_O})$ is produced with the owner's identity watermark W_O. Finally, $E_{pk_O}(X_{W_O})$ is sent to the owner. After receiving $E_{pk_O}(X_{W_O})$, the owner gets the plain image $X_{W_O}$with his/her private key sk_O.

The image retrieving process prevents the image owner from obtaining the same image as the one that the cloud server stores. The retrieved copy from the cloud server can be traced through

watermark extraction. The extracted watermark W_O means that this copy has been taken back by the owner. Though the owner deliberately spreads the retrieved copy, he/she can not charge the innocent CSP.

In the following, we will elaborate the details of the CUW protocol. For clarification, the whole protocol is divided into three parts. At first, we describe the watermarking protocol. Next, the owner identification protocol is presented. Lastly, the arbitration protocol is exhibited.

4.2. Watermarking Protocol

In this subsection, we describe the details in the watermarking protocol, including watermark generation and embedding. The CWU protocol defines three types of watermark, an ownership watermark O_W, a CSP's identity watermark W_C and an owner's identity watermark W_O. O_W is generated by the true image owner, used for claiming the ownership of his/her image, but kept by WCA. Each W_C is associated with each specific CSP. WCA generates and embeds W_C into an owner's image before it being outsourced to a cloud server. We employ W_C to detect and trace a certain CSP. Similar to W_C, each owner's identity watermark W_O is linked to each specific image owner. W_O is embedded into an image by WCA when an owner asks to take the image back.

For simplicity, we denote the image owner as *O* and the cloud server as *C*. Interactions among *O*, *C* and WCA depend on the underlying watermarking and homomorphic encryption techniques. Fig. 2(a) and Fig. 2(b) show the image outsourcing and retrieval interactive steps, respectively.

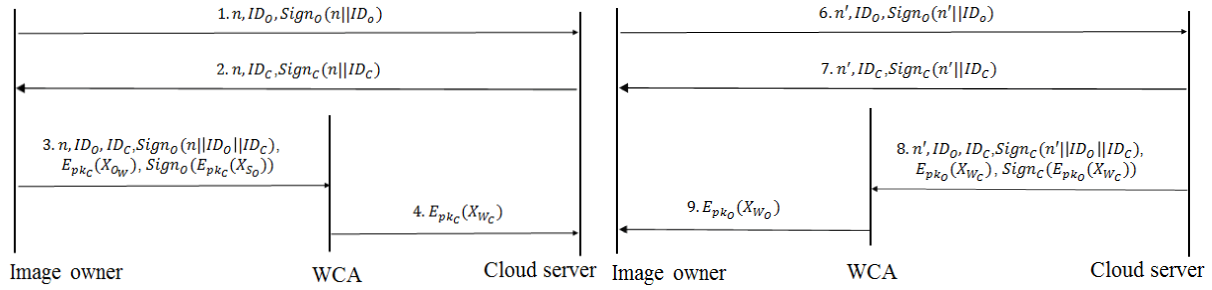


Fig. 2. Details of the proposed protocol. (a) The image outsourcing process; (b) The image retrieval process

4.1.1. The image outsourcing process

Step 1. *O* generates a new odd number *n*, implying a request of uploading an image to *C*. Then, O sends *n*, *O*'s identity ID_O and $Sign_O(n||ID_O)$ to *C*.

Step 2. When *C* receives *n*, ID_O and $Sign_O(n||ID_O)$, *C* verifies the validity of the signature, and aborts the image upload if it is invalid. Otherwise, *C* forwards *n*, *C*'s identity ID_C and $Sign_C(n||ID_C)$ to *O*.

Step 3. Upon receiving *n*, ID_C and $Sign_C(n||ID_C)$ from *C*, *O* checks the validity of the signature, and returns to the Step 1 if it is invalid. Otherwise, he/she inserts his/her ownership watermark O_W into the image *X* and gets the plain watermarked image X_{O_W} . Note that in this step, *O* is free to use any robust watermarking scheme to resist various attacks. O_W is solely to assist O to prove his/her ownership of *X*. Afterward, *O* computes $E_{pk_C}(X_{O_W})$ and conveys *n*, ID_O, ID_C, O_W, $E_{pk_C}(X_{O_W})$, $Sign_O(n||ID_O||ID_C)$ and $Sign_O(E_{pk_C}(X_{O_W}))$ to WCA.

Step 4. After WCA receives $Sign_O(n||ID_O||ID_C)$ and $Sign_O(E_{pk_C}(X_{O_W}))$ from *O*, WCA verifies the signatures, and terminates the image upload if any of them is invalid. Otherwise, WCA generates a valid watermark W_C specific to the CSP. Then, WCA computes the results according to equation (6) in the encrypted domain.

$$E_{pk_C}(X_{W_C}) = E_{pk_C}(X_{O_W} \oplus W_C) = E_{pk_C}(X_{O_W}) \oplus E_{pk_C}(W_C), \quad (6)$$

where $E_{pk}(\cdot)$ is homomorphic with respect to $\oplus$. Note that WCA has no access to the plain image X_{O_W}. Next, WCA forwards $E_{pk_C}(X_{W_C})$ and $Sign_{WCA}(E_{pk_C}(X_{W_C}))$ to *C* in terms of ID_C. Lastly, WCA stores *n*, ID_O, ID_C, O_W, $Sign_O(n||ID_O||ID_C)$ and W_C in the *n-th* entry of $Table_{ID_O}$.

Step 5. Receiving $E_{pk_C}(X_{W_C})$ and $Sign_{WCA}(E_{pk_C}(X_{W_C}))$, *C* verifies the integrity of the signature. If the signature is invalid, *C* sends a transmission failure signal to WCA and WCA returns Step 4. Otherwise, *C* obtains X_{W_C} by computing equation (7) and stores X_{W_C} in the cloud storage. Moreover, *C* hopelessly erases or substitutes the embedded watermark W_C with the absence of W_C.

$$X_{W_C} = D_{sk_C}(E_{pk_C}(X_{W_C})) \quad (7)$$

4.1.2. The image retrieval process

Step 6. *O* sends the even number $n' = n + 1$, implying a request of downloading the image X_{W_C} from *C*. Then, *O* sends n', ID_O and $Sign_O(n'||ID_O)$ to *C*.

Step 7. Upon receiving n', ID_O and $Sign_O(n'||ID_O)$, *C* verifies the validity of the signature, and aborts the image download operation if it is invalid. *C* answers *O* n', ID_C and $Sign_C(n'||ID_C)$. Meanwhile, *C* encrypts the image X_{W_C} with the owner's public key pk_O and gets $E_{pk_O}(X_{W_C})$. Subsequently, *C* forwards n', ID_O, ID_C, $E_{pk_O}(X_{W_C})$, $Sign_C(n'||ID_O||ID_C)$ and $Sign_C(E_{pk_O}(X_{W_C}))$ to WCA.

Step 8. After receiving the message of n', ID_O, ID_C, $E_{pk_O}(X_{W_C})$, $Sign_C(n'||ID_O||ID_C)$ and $Sign_C(E_{pk_O}(X_{W_C}))$, WCA verifies the signatures, and ends the image upload if any of them is invalid. Otherwise, WCA generates a valid watermark W_O, relative to the specific image owner. Then, WCA computes the results according to equation (8), adopting the properties of homomorphic encryption.

$$E_{pk_O}(X_{W_O}) = E_{pk_O}(X_{W_C} \oplus W_O) = E_{pk_O}(X_{W_C}) \oplus E_{pk_O}(W_O) \quad (8)$$

Next, WCA sends $E_{pk_O}(X_{W_O})$ and $Sign_{WCA}(E_{pk_O}(X_{W_O}))$ to *O* in terms of $ID_O$and adds the new item W_O into the *n-th* entry of $Table_{ID_O}$.

Step 9. Receiving $E_{pk_O}(X_{W_O})$ and $Sign_{WCA}(E_{pk_O}(X_{W_O}))$ from WCA, *O* verifies the integrity of the signature. If the signature is invalid, *O* conveys a transit failure signal to WCA and WCA returns Step 8. Otherwise, *O* decrypts the ciphertext image $E_{pk_O}(X_{W_O})$ and obtains the image X_{W_O}, with his/her private key sk_O by computing equation (9).

$$X_{W_O} = D_{sk_O}(E_{pk_O}(X_{W_O})) \quad (9)$$

It is unworkable to remove the watermark X_{W_O} because *O* lacks for the knowledge of it.

Step 10. If O wants to delete his/her image, stored in the cloud server of ID_C, he/she sends the deletion request $Deletion_O$, ID_O and ID_C to WCA. Then, WCA adds the new item $Deletion_O$ into the *n-th* entry of $Table_{ID_O}$. Finally, WCA transmits $Deletion_O$ and ID_O to the cloud server as the deletion signal. The cloud server should carry on the entire removal of the *n-th* image related to the owner of ID_O.

4.3. Owner Identification Protocol

When an illegal copy Y of the image X, which should have been entirely deleted by a CSP's cloud server, is discovered, the image owner can execute the violator identification protocol described in this subsection and identify the unlawful CSP.

The image owner first does the corresponding watermark extraction algorithm on Y. The extracted ownership watermark is denoted as $O_{W'}$. Then, ID_O and $O_{W'}$ are sent to WCA. After that, WCA correlates $O_{W'}$ with O_W stored in $Table_{ID_O}$ and calculates the correlation value. If the value is beyond the predetermined threshold, WCA believes that the suspected image owner is the true image owner of Y. WCA collects the associated information stored in the matched entry and caries on the arbitration protocol. Otherwise, the owner identification protocol ends in failure.

4.4. Arbitration Protocol

If the owner identification protocol succeeds, WCA fetches the matched *n-th* entry in $Table_{ID_O}$. With the deletion request $Deletion_O$ recorded in this entry, WCA checks whether W_C exists in Y. If the item W_O is also kept in the table, WCA has to check the existence of W_O, too. Because there is a special scenario, where the owner has taken Y back from the cloud server. If W_C is indeed found in Y without W_O, the associated CSP is guilty of infringing the owner's right to be forgotten. If W_O is also found, WCA has reason to believe that Y has been retrieved from the cloud server. The cloud server disables access to the retrieved image. Thus, the relative CSP is innocent.

5. SECURITY ANALYSIS

In the proposed protocol, anyone among the image owner, the cloud server and WCA can strike security problems. In addition, the security of the CUW protocol relies critically on the security of the underlying encryption and watermarking techniques. This section describes the security analysis of main issues and storage consumption.

1) Data Privacy

- The image owner has access to the original image X, the watermarked copy X_{O_W} containing the ownership watermark O_W and the retrieved image X_{W_O} containing his/her identify watermark W_O. It is the owner's business to decide whether or not expose X, X_{O_W} or X_{W_O}.
- According to our protocol, WCA only receives the encrypted image $E_{pk_C}(X_{O_W})$ and computes $E_{pk_C}(X_{W_C})$ and $E_{pk_O}(X_{W_O})$ under homomorphic encryption. Thus, plain images will not be leaked from WCA.
- The cloud server can only touch the plain image X_{W_C}, which contains its watermark W_C. Once the cloud server reveals X_{W_C}, it is easily feasible to trace its misbehavior. Hence, the privacy of plain images in the proposed protocol is perfectly maintained.

2) **Traceability**

It is technically feasible to distinguish the dishonest party through detecting the embedded identity watermarks. The robust watermarking scheme is the cornerstone of successful traceability.

3) **Non-Frameability**

- An image owner can only touch and wilfully disseminate plain images X, X_{O_W} and X_{W_O}. It is obvious that no CSP's identity watermark W_C is embedded in X or X_{O_W}. Besides, X_{W_O} both contains W_C and W_O. To frame the CSP, one way is that the owner embeds W_C in X_{O_W} and gets X_{W_C}. However, W_C is kept secret solely by WCA. Another way is that the owner erases W_O from X_{W_O} and obtains X_{W_C}. Even though the owner may get all positions of embedding W_C and W_O by comparing X_{O_W} and X_{W_O}, it is difficult for the owner to tell which positions are used to embed W_C and which positions are used to embed W_O.
- CSP can merely obtain plain image X_{W_C}without the owner's unique watermark W_O. It is difficult for CSP to forge the image X_{W_O}.

In a word, the proposed protocol is secure and fair because both the image owner and the CSP discourage to charge each other without the other side's identity watermark.

4) **The Right to Be Forgotten**

Another practical application of the proposed protocol is to protect the owner's right to be forgotten. An image owner has sent his/her deletion request of the image X_{W_C}. But the owner later meets the copy of the image X_{W_C}. Then the owner can employ the owner identification and arbitration protocol and lastly can indict the dishonest CSP.

5) **Storage Requirements**

- The image owner outsources his/her local images to cloud servers. In our scheme, after uploading images, the owner may reserve secret keys to control embedding positions of his/her ownership watermark. This depends on the underlying watermark scheme.
- WCA has certain storage space to keep the records of watermark for the owner identification part and arbitration part. The size of the record table grows linearly with the number of cloud users.
- CSP offers large storage space as service, so it is reasonable to put the burden of storing outsourced images.

6. WATERMARK EMBEDDING

In order to construct a practical system under the proposed protocol, we employ two kinds of concrete watermark schemes as examples. Firstly, we present the identity watermark scheme for embedding W_C and W_O. Next, the ownership watermark scheme is added to the whole water-marking protocol.

6.1. Identity Watermark Scheme

In the constructed system, the primary watermarking technique has to be combined with the homomorphic cryptosystem. Thus, we describe the corresponding changes in detail.

6.1.1. Dither Modulation

The value of the to be quantized coefficient x may be a float number. However, the applied homomorphic cryptosystem is based on the algebraic property of the integer number. Thus, we scale all to be quantized coefficients with a constant factor s before encryption. By assuming an

additively homomorphic cryptosystem, the scaled equation (5) can be translated into the encrypted domain as

$$E(y) = \begin{cases} E(\lceil s(Q_{\Delta-even}(x+d)-d)\rfloor) \times E(w)^{\Delta^s}, & if\ w=0, \\ E(\lceil s(Q_{\Delta-odd}(x+d)-d)\rfloor) \times E(w)^{\Delta^s}, & if\ w=1, \end{cases} \tag{10}$$

where $\lceil \cdot \rfloor$ is the rounding function. However, the transform coefficients of an image may be negative, and we need to consider the problem of representing the negative integers in the cryptosystem. Suppose *N* is the modulus of the cryptosystem. We let $N \geq 2sup\{\lceil s(Q_\Delta(x+d)-d)\rfloor\}+1$, where $sup\{\cdot\}$ denotes the least upper bound operator performed on the coefficients. After decryption, the watermarked host can be obtained through divided by the scaling factor *s*.

In the system, WCA receives the encrypted coefficient $E(\lceil s(Q_\Delta(x+d)-d)\rfloor)$, but does not know the corresponding parity of the plain value. In such a situation, WCA can not embed the watermark bit *w* by checking the parity of $E(\lceil s(Q_\Delta(x+d)-d)\rfloor)$. Therefore, we propose that the coefficient is only quantized to the nearest even multiples of Δ. The ultima resulting embedding equation can be summarized as

$$E(y) = E(\lceil s(Q_{\Delta-even}(x+d)-d)\rfloor) \times E(w)^{\Delta^s} \tag{11}$$

6.1.2. Quantization Table

Generally speaking, the transformed coefficients are more resilient against various attacks and signal processing. Here, the Discrete Cosine Transform (DCT) is applied to hide data in a fixed low frequency band, and the DC coefficient should avoid being utilized for minimal visual distortion. For good perceptual quality of an image, we use the 8×8 quantization table M^q from JPEG compression algorithm and use the quality factor q to control the quantization step size, denoted as $M^q_{i,j}$, where $i,j \in \{1,\dots,7\}$.

6.1.3. Embedding Procedure

In the image outsourcing process, the embedding procedure of the CSP's W_C identity watermark is performed as the following steps:

Step 1. An image is partitioned into 8×8 non-overlapping blocks. Using the owner's secret key *key*, several selected blocks are specified to carry the ownership watermark as mentioned later.

Step 2. After embedding the ownership watermark, the owner uses key to choose another blocks, each of which is transformed by DCT. Denote the DCT coefficients of each block as $c_{i,j}$ $(0 \leq i,j \leq 7)$. Several DCT coefficients $c_{i,j}$ of a block in the fixed low frequency band are quantized to the nearest even number by quantizing step size $M^q_{i,j}$.

$$\overline{c_{i,j}} = Q_{M^q_{i,j}-even}(c_{i,j}+d_{i,j}) - d_{i,j}, \tag{12}$$

where $d_{i,j}$ represents the dither value indexed by *i*, *j* in a block. The other DCT coefficients are never selected for embedding and are quantized to the nearest integer as

$$\overline{c_{i,j}} = Q_{M^q_{i,j}}(c_{i,j}+d_{i,j}) - d_{i,j} \tag{13}$$

Then scale all quantized $\overline{c_{i,j}}$ with a constant factor *s*, and get $\overline{sc_{i,j}} = \lceil s \times \overline{c_{i,j}} \rfloor$.

Step 3. Each scaled and quantized coefficient is encrypted using the CSP's public key pk_C. The encrypted coefficient is $E_{pk_C}(\overline{sc_{i,j}})$. After all blocks have been processed, the image owner

obtains the entire encrypted DCT coefficients, and then sends these coefficients to WCA. At the same time, the image owner sends his/her key *key*, the quality factor q and the scale factor s to WCA.

Step 4. Upon receiving the encrypted DCT coefficients, q, and s from the image owner, WCA generates the unique CSP's identity watermark W_C, encrypts it with pk_C, and uses *key* to select the fixed encrypted DCT coefficients. Then, WCA executes the bit-by-bit embedding operation as equation (11).

Step 5. After all encrypted watermarked DCT coefficients have been obtained, WCA sends these coefficients to the CSP's cloud server along with the scale factor s.

Step 6. When received the encrypted DCT coefficients from WCA, the cloud server first decrypts them, rescales the DCT coefficients through dividing by the scale factor s, and then performs the inverse discrete cosine transform (IDCT). Lately, the watermarked plain image is obtained.

In the image retrieval process, the embedding steps of the image owners identity watermark W_O are performed as alike as Step 2 to Step 6 in the image outsourcing process. Note that no ownership watermark O_W is embedded in the image retrieval process. Here, the embedding positions of W_C and W_O should not overlap those of O_W. But it does not matter whether the embedding positions of W_O overlap those of W_C. For simplicity, we only consider that the embedding positions of the two identity watermarks do not overlap.

6.1.4. Extraction Procedure

The extracted watermarks are required for the owner identification protocol and arbitration protocol. After the image X outsourced, the image owner finds an illegal spread copy Y of X. The owner partitions Y into non-overlapping blocks and then, extracts the watermark O_W for the certification of ownership. If his/her ownership of Y is confirmed, WCA will extract identity watermarks as follows.

WCA uses the secret key *key* to locate the embedding blocks. These blocks are performed DCT and the fixed low coefficients are quantized using the corresponding quantization step sizes. Then WCA checks the odd-even property of the quantized coefficient value. If the value is even, the extracted watermark bit is regarded as 0, otherwise 1.

6.1.5. The Number of Watermarked Blocks

The number of the identity watermark bits is denoted as N_{IW}, and the number of embedded bits in each block is denoted as N_{EB}. Then, the number of watermarked blocks, denoted as N_{WB}, can be calculate as

$$N_{WB} = \frac{N_{IW}}{N_{EB}} \tag{14}$$

6.2. Ownership Watermark Scheme

The ownership watermark O_W is embedded in the original image by the image owner before the image is sent to a cloud server. When an illegal copy appears, the ownership watermark is recovered and then the extraction of the identity watermark is executed. It is also important to employ the robust watermark scheme in the owner identification process. In order to reduce the distortion of host images, we use the second generation watermark (Kutter, Bhattacharjee, & Ebrahimi, 1999) as the embedding scheme of O_W. Second generation watermark schemes almost

do not affect the host quality and can enhance robustness of the embedded watermark, as the watermark is associated with robust features of the host.

Here, we employ a simple watermarking scheme based on wavelet decomposition. Before outsourcing the image, the image owner chooses a unique binary sequence $W = \{w(i)|w(i) = \{0,1\}, 0 \leq i \leq L-1\}$ of of length L as the ownership watermark O_W and selects the non-overlapping blocks using his/her secret key *key*. A new sub-image, consisting of these blocks, is denoted as *subI*.

6.2.1. Embedding Procedure

Before an image outsourced, the ownership watermark O_W is embedded into the image as the following steps:

Step 1. Perform three wavelet decompositions on *subI* and obtain the approximate subgraph coefficients LL_3.

Step 2. The watermark *W* is embedded into the LL_3 using the superposition method as

$$LL_3' = LL_3 + \alpha W \tag{15}$$

Step 3. Apply the Discrete Fourier Transform (DFT) on the modified band LL_3' and get its DFT coefficients *f*. The DFT coefficients are complex data and can be written as $f = a + bi$, *a* is the real par and *b* is the imaginary part. Take the low and middle frequency coefficients from all the DFT coefficients and compute a feature vector $V = \{v(i)|v(i) = \{0,1\}, 0 \leq i \leq L-1\}$ by the symbolic computation function as

$$v(i) = \frac{sign(a)+1}{2} \tag{16}$$

$$v(i+1) = \frac{sign(b)+1}{2} \tag{17}$$

where $sign(x) = x/|x|$.

Step 4. The binary sequence *B* is generated by xor the feature vector *V* and the watermark *W* as $B = V \oplus W$. The image owner sends *W* and *B* to WCA for the image ownership authentication.

Step5. Use LL_3' and other sub-bands to reconstruct the new watermarked subimage through the wavelet reconstruction.

6.2.2. Extraction Procedure

If the image owner finds that an illegal copy *Y* seems to be his/her deleted image *X*. He/She can extract the ownership watermark for the assurance of his/her ownership. Step 1 in the extracting procedure is the same to Step 1 in the embedding procedure. The feature vector $V' = \{v'(i)|v'(i) = \{0,1\}, 0 \leq i \leq L-1\}$ is computed by the operation as Step 4 in the embedding procedure. Then, the owner asks WCA for *W* and *B*. Lastly, the watermark W' is extracted by $W' = V' \oplus B$. If the correlation between *W* and W' is greater than the predefined threshold, the doubtful owner is considered to be the true owner of *Y*.

6.2.3. The Number of Watermarked Blocks

The number of the ownership watermark bits is N_{OW}. Then, the number of watermarked blocks, denoted as N_{WB}, can be calculated as

$$N_{WB} = N_{OW} \tag{18}$$

7. PERFORMANCE EVALUATION

The proposed protocol is based on the ownership and identity watermark schemes. The performance of the two watermark schemes helps to evaluate the performance of our constructed system, including the visual quality of users' images, the success probability of finding an illegal CSP and the system capacity. The details will be exhibited in the following subsection.

7.1. Two Watermark Schemes

Invisibility and robustness of watermarked images are the two types of important performance. The visual quality of watermarked images is measured by the peak signal to noise ratio (*PSNR*) while the robustness of watermarking schemes is precisely evaluated by the bit correct ratio (*BCR*). Higher *PSNR* indicates better perceived quality and larger *BCR* suggests stronger robustness. Here, the test image library consists of 1000 distinct gray images with size of 512×512.

7.1.1. Ownership Watermark Scheme

In the section, the whole host image is used for embedding the ownership watermark. In the ownership watermark scheme, the embedding strength α of watermark is considered as the influence factor. The average *PSNR* and average *BCR* of the 1000 images under different αs are shown in Table 2.

Table 2. PSNR and BCR under different αs in the ownership watermark scheme

α					
0.01		0.05		0.1	
PSNR	*BCR*	*PSNR*	*BCR*	*PSNR*	*BCR*
108.94	1.00	94.30	0.999	88.91	0.998

From Table 2, we can see that the ownership watermark scheme, which can achieve high *PSNR*s, almost has no visual impact on host images. Additionally, strong robustness can also be achieved for large *BCR*s.

7.1.2. Identity Watermark Scheme

As for the identity watermark scheme, we take into account two influence factors, the number of embedded bits in each 8×8 block N_{EB} and the quality factor q. Here, each block of each image is used to embed the identity watermark. Before encryption, the coefficients are multiplied by $s = 2^{16}$ as in the equation (11). At q = 55, the average *PSNR* and average *BCR* of the 1000 images are recorded in Table 3 for various N_{EB}s.

Table 3. PSNR and BCR under different N_{EB}s at q = 55 in the identity watermark scheme

N_{EB}							
1		2		3		4	
PSNR	*BCR*	*PSNR*	*BCR*	*PSNR*	*BCR*	*PSNR*	*BCR*
34.39	0.73	31.81	0.97	31.72	0.97	31.68	0.97

From Table 3, we can find that with N_{EB} increasing, the visual quality of images degrades slightly while the robustness first improves dramatically and then stays stable.

In the identity watermark scheme, there is a tradeoff, controlled by q, between the robustness and perceptual quality. In order to highlight this tradeoff, we draw average *PSNR* and average *BCR* curves at N_{EB} = 2 in Fig. 3. Fig. 3 shows that the watermarked image quality can be improved by the increased q while robustness against attack is decreased.

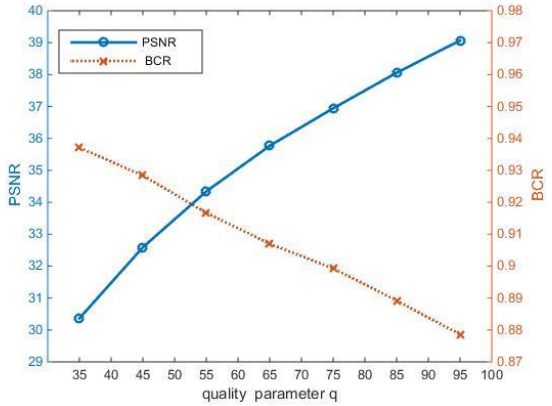


Fig. 3. PSNR and BCR with different qs

7.2. Watermarks in the Proposed Protocol

Two properties, including the visual quality of users' images, denoted as VP_u, and the success probability of finding an illegal CSP, denoted as P_{suc}, are estimated for testing the availability of our proposed protocol. *PSNR*s of the ownership and identity watermark schemes help to evaluate the VP_u of users' images. In the CUW protocol, it can be allowed to extract the identify watermark only after the owner watermark is successfully extracted. Thus, we use the multiplication of *BCR*s of the two watermark schemes as P_{suc}.

Each image is divided into 8×8 non-overlapping blocks. Half of these blocks are randomly selected to embed the ownership watermark, and the other half to embed the identity watermark. For simplicity, the average *BCR* of ownership watermarks is denoted as BCR_O while the average BCR of identity watermarks is denoted as BCR_I.

The performance of our constructed system is also studied when these 1000 images suffer common attacks, such as JPEG Compression, additive white Gaussian noise (AWGN), gaussian low pass filtering, image rescaling and rotation. The JPEG compression strength is controlled by the quality factor QF and the variance of AWGN is denoted as σ_n^2 . The window size of gaussian low pass filter is 3 × 3, the rescaling factor of 0.8, and the rotation angle of 10°. When α= 0.05, q = 55 and N_{EB} = 2, the results in different cases are listed in Table 4.

Table 4. Watermark tolerance in the proposed protocol under different attacks

		VP_u	BCR_O	BCR_I	P_{suc}
No attack		40.33	1.0	0.91	0.910
QF	45	33.29	0.995	0.95	0.945
	65	35.14	0.996	0.92	0.916
σ_n^2	9	34.31	0.996	0.81	0.807
	16	36.78	0.997	0.90	0.897

Gaussian low pass filtering	3 × 3	36.83	0.994	0.600	0.569
Rescaling	0.8	37.49	0.984	0.481	0.473
Rotation	10	30.743	0.962	0.665	0.639

From Table 4, we can find that under no attack the average VP_u is over 30 dB so that visual quality of watermarked images is acceptable. Under JPEG Compression, BCR_Os and BCR_Is are more than 90% and finally, P_{suc}s are more than 90% with slight changes. BCR_Is under AWGN decrease and P_{suc}s decrease but still beyond 80%. Under gaussian low pass filter, rescaling and rotation, BCR_Is are low, leading to quite low P_{suc}s.

It is seen from these figures that the success probability P_{suc} of tracing back to an illegal leaker relies on high BCR_O and high BCR_I. Low BCR_O or low BCR_I makes the trace failure. Besides, note that the test parameters, like N_{EB} used here, are not claimed to be the optimal ones. One can adjust the parameters flexibly according to the application scenarios.

7.3. System Capacity

In this subsection, we consider another property of the CUW protocol, the system capacity. The system capacity means how many cloud users and how many CSPs that can be accommodated. We denote the number of cloud users as *NU* and the number of CSPs as *NC*.

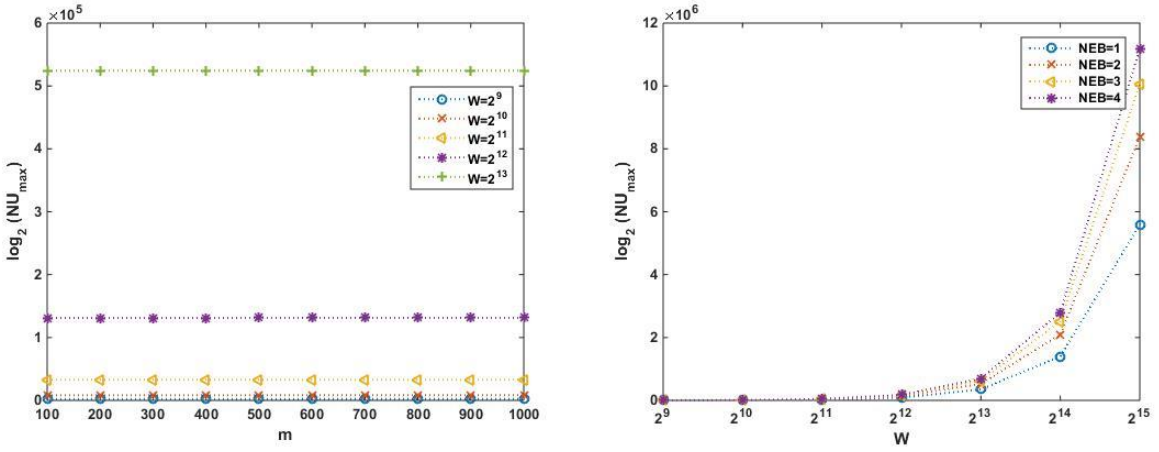
Since the ownership watermark O_W and the image owner's identity watermark W_O both serve the image owner (the cloud user), the lengths of the two binary watermarks are equivalent to $\lceil log_2(NU) \rceil$. The length of the CSP's identity watermark W_C can be expressed as $\lceil log_2(NC) \rceil$. A host image sizes $W \times H$. According to the two watermark schemes, the following formula is derived:

$$\lceil log_2(NU) \rceil + \frac{\lceil log_2(NU) \rceil}{N_{EB}} + \frac{\lceil log_2(NC) \rceil}{N_{EB}} \leq \frac{W \times H}{8 \times 8} \tag{19}$$

In the real application, *NU* is far more than *NC*. Thus, we investigate more on *NU* and we assume that $NC = \frac{1}{2^m} NU$. Then, the above formula can be expressed as

$$\lceil log_2(NU) \rceil + \frac{\lceil log_2(NU) \rceil}{N_{EB}} - \frac{m}{N_{EB}} \leq \frac{W \times H}{8 \times 8} \tag{20}$$

When N_{EB} = 2 and $W = H$, the max *NU* against *m* in logarithmic scale is drawn in Fig. 4(a) with different *W*s. When m = 1000 and N_{EB} s from 1 to 4, the logarithmic curves of max *NU* against *W* are drawn in Fig. 4(b).



(a) *(b)*

Fig. 4. System capacity of the proposed protocol. (a) with different ms and Ws; (b) with different Ws and N_{EB} s

Fig. 4(a) shows that *m* almost has no impact on the max *NU* even though *m* is more than 100. However, from Fig. 4(b), we can see that with *W* increasing, the max *NU* grows rapidly. Compared with *m*, *W* affects more on the system capacity. Additionally, the rising N_{EB} can improve *NU* and enhance the system capacity. In the real world, according to the system requirements, the corresponding error correction coding can be adopted, which will reduce the system capacity. Generally, stronger resistance to attacks consumes more system capacity.

8. DISCUSSION

This paper studies the problem of deleting plain data assuredly from cloud servers. That is, when a cloud user deletes his/her data from in-the-wild cloud servers, there is no guarantee that the CSP will completely delete the data from its servers. This paper presents the CUW protocol between cloud users and CSPs. But there are some limitations of the proposed protocol.

One of the limitations is that the CUW protocol introduces the third party WCA. WCA will lead to extra economic cost. The second limitation is that homomorphic encryption consumes a lot of time, hindering the quick and timely application. Lastly, the multimedia data downloaded by a cloud user from the cloud server will include two identifier watermarks. The quality of the host data is destroyed again.

We demonstrate the feasibility of the CUW protocol just using the off-the-shelve cryptography algorithm and watermarking algorithms. But it is not limited to these algorithms. With the development of ciphertext computing and watermark techniques, more quick cryptography and more robust watermarking algorithms can be applied to the CUW protocol.

9. CONCLUSIONS

In this paper, we propose a cloud-user watermarking protocol, which provides the first technical approach to solving the right to be forgotten problem. The CSP against this right can be found with technical evidence. Besides, the proposed protocol supports assured deletion of plain data in the cloud storage environment. We implement an image system under our protocol framework and demonstrates its availability and practicality.

As future work, there are still some aspects that can be improved. First, we will design an anonymous protocol to protect the user's privacy, and maintain the real users' right to be forgotten. Second, the watermark protocol will be modified such that the embedded watermarks can be erased. Then, the original data can be recovered from watermarked copies, and cloud users can take back lossless data.

10. ACKNOWLEDGEMENTS

This work was supported in part by the Natural Science Foundation of China under Grant U1636201, 61572452.

REFERENCES

Chen, B., & Wornell, G. W. (2001). Quantization Index Modulation: A Class of Provably Good

Methods for Digital Watermarking and Information Embedding. *IEEE Transactions on Information Theory*, 47(4), 1423-1443.
Damgård, I., & Jurik, M. (2001). A Generalisation, a Simpli.cation and Some Applications of Paillier's Probabilistic Public-Key System. *Lecture Notes in Computer Science*, 7(45), 119-136.
Europea, U. (1995). Directive no. 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Luxembourg.
Frattolillo, F. (2017). Watermarking protocols: an excursus to motivate a new approach. *International Journal of Information Security* (4), 1-15.
Jin, H. (2009). Efficient Forensic Analysis for Anonymous Attack in Secure Content Distribution. *International Journal of Digital Crime and Forensics* (IJDCF), 1(1), 59-74.
Kropf, J. W. (2014). Google Spain SL v. Agencia Española de Protecci´on de Datos (AEPD). Case C-131/12. *American Journal of International Law*, 108(3), 502-509.
Kuribayashi, M., & Tanaka, H. (2005). Fingerprinting Protocol for Images Based on Additive Homomorphic Property. *IEEE Transactions on Image Processing*, 14(12), 2129-2139.
Kutter, M., Bhattacharjee, S. K., & Ebrahimi, T. (1999). Towards Second Generation Watermarking Schemes. *International Conference on Image Processing*, 1999. ICIP 99. Proceedings (Vol.1, pp.320-323 vol.1). IEEE.
Memon, N., & Wong, P. W. (2001). A Buyer-Seller Watermarking Protocol. *IEEE Transactions on Image Processing*, 10(4), 643-649.
Paillier, P. (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. *International Conference on Theory and Application of Cryptographic Techniques* (Vol.547, pp.223-238). Springer-Verlag.
Perlman, R. (2005). File System Design with Assured Delete. *IEEE International Security in Storage Workshop* (pp.6-88). IEEE.
Priebe, C., Muthukumaran, D., O'Keeffe, D., Eyers, D., Shand, B., Kapitza, R., & Pietzuch, P. (2014). Cloudsafetynet: Detecting Data Leakage Between Cloud Tenants. In *Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security* (pp. 117-128). ACM.
Prins, J. P., Erkin, Z., & Lagendijk, R. L. (2007). Anonymous Fingerprinting with Robust QIM Watermarking Techniques. *Eurasip Journal on Information Security*, 2007(1), 1-13.
Ramokapane, K. M., Rashid, A., & Such, J. M. (2016). Assured Deletion in the Cloud: Requirements, Challenges and Future Directions. *ACM on Cloud Computing Security Workshop* (pp.97-108). ACM.
Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation*, 169-179.
Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of the ACM*, 21(2), 120-126.
Stinson, D. R. (2005). *Cryptography: theory and practice*. CRC press.
Tang, Y., Lee, P. P. C., Lui, J. C. S., & Perlman, R. (2012). Secure Overlay Cloud Storage with Access Control and Assured Deletion. *IEEE Transactions on Dependable and Secure Computing*, 9(6), 903-916.
Vrable, M., Savage, S., & Voelker, G. M. (2009). Cumulus: Filesystem Backup to the Cloud. *ACM Transactions on Storage (TOS)*, 5(4), 14.
Xia, Z., Wang, X., Zhang, L., Qin, Z., Sun, X., & Ren, K. (2016). A Privacy-Preserving and Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing. *IEEE Transactions on Information Forensics and Security*, 11(11), 2594-2608.