Robust Image Document Authentication Code with Autonomous Biometric Key Generation, Selection, and Updating in Cloud Environment

Zaid Ameen Abduljabbar[1,2], Hai Jin[1], Zaid Alaa Hussien[1,3], Ali A.Yassin[2],
Mohammed Abdulridha Hussain[1,2], Salah H. Abbdal[1], Deqing Zou[1]
[1]Cluster and Grid Computing Lab, Services Computing Technology and System Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China
Email: zaidalsulami@yahoo.com, hjin@hust.edu.cn
[2]University of Basrah, Basrah, Iraq. [3]Southern Technical University, Basrah, Iraq

***Abstract*—Recently, security issues are obstructing the development and using of cloud computing services. Authentication and integrity play an important role in the cloud security, and numerous concerns have been raised to recognize any tampering with exchanges of the image document between two entities (sender and receiver) within the cloud environment. However, none of the existing solutions reduce the probability of known attacks by combining cryptographic hash function with a strong factor that should be periodically changed. For this reason, in this paper we propose a robust one-time image document authentication scheme based on combining non-interactive one-time biometric key and a robust wavelet-based cryptographic hashing scheme. The result of the combination is one-time image document authentication code (*OMAC*). *OMAC* is hidden in an image document as a cover image through reversible data embedding steganography. The proposed scheme has several important security attributes, such as key agreement, biometric key management, robust *OMAC*, invulnerability, and efficiency. In biometric key management, key generation, key selection, and key update algorithms are performed autonomously by the sender and the receiver; thus, no interaction between them is needed.**

***Keywords*—Biometric key management; cloud computing; OMAC; session key agreement**

I. Introduction

A paper document is an essential communication channel and a form of keeping information. Documents, such as bank checks, passports, identity cards, financial instruments, legal documents, licenses, birth certificates, digital books, maps, engineering, architectural fees, and road maps, are exchanged and disseminated daily at high volume. Nowadays, the transmission and storage of electronic information cost less. Thus, electronic documents are more commonly used than printed documents, and they have radically increased and are exponentially distributed [1]. Cloud computing is generally regarded as the next-generation computing infrastructure because of its effectiveness in enabling users to utilize abundant resources and its efficient and readily available on-demand service [2]. Consequently, users of cloud computing have raised information security requirements for their communication compared with other challenges [3]. Image document integrity and origin achieved through communication between two endpoints (sender and receiver) have become huge cloud security challenges.

In recent years, a number of authors have proposed different authentication and integrity schemes based on cryptographic hash function to generate a robust hashed value and transfer it securely. In 2008, Rabadi and Mahmud [4] presented the concept of message anonymity. In their scheme, the message authentication protocol uses *message authentication code* (MAC) from vehicle to vehicle to provide anonymity, authentication, and message integrity. The concept of MAC anonymity depends on a timestamp, which is a one-time factor to generate an anonymous message. However, this scheme entailed additional costs because an extra hardware device to save the ID and shared symmetric secret was required on each vehicle. Moreover, the security analysis of the proposed protocol was not discussed. Later in 2010, Jamil and Aziz [5] proposed the concept of permutation key to form a transformed image. The authors proposed a secure hashing scheme between two entities to overcome the problem of security over communication. They used a permutation key for each block of the divided image to make the hashed value much harder to be guessed by an attacker. The weakness of this scheme is that the permutation key is computed based on feature extraction to generate the image hash. Thus, an attacker might know this and recompute the permutation key to regenerate the hashed value.

In 2011, the concept of message anonymity has returned again by Liu et al. [6], they suggested a hash-based secure interface between two entities over the Internet. A one-time shared private key, a public hash function, a timestamp, and a validity period were utilized to generate one-time message anonymity. The types of attack that this scheme could withstand were not clarified. In the same year, Naqvi and Akram [7] suggested a concept of increasing the robustness of the key-based hash message authentication code (HMAC-MD5). They proposed the use of the MD6 compression function to generate a robust key to compute HMAC. The key was generated by MD6 maintained randomization and was difficult for attackers to predict. However, the security analysis of the proposed scheme was not discussed extensively and was limited to birthday and exhaustive key search attacks.

978-1-4673-8715-6/15/$31.00 ©2015 IEEE

In 2012, Chaisri et al [8] proposed the concept of maintaining the integrity of a faxed document. They suggested a scheme to protect the integrity of a faxed document by using *data encryption standard* (DES) to encrypt and protect the hashed value MAC-MD5. One weakness of this scheme is that the sender needs to add the secret key of DES to the faxed document before sending to the receiver. Thus, an attacker could extract and reuse the secret key to decrypt the MAC. Furthermore, an attacker could reuse the secret key to regenerate a fake faxed document with the receiver believing that it originated from a valid sender.

Recently, in [9] Shen and Liu developed a content-based watermarking method based on the Markov chain to protect the digital document and images authentication code. The weakness of this scheme is that the embedded authentication code is applied based on sequence mapping in the *least significant bits* (LSBs) of a cover-image, which lacked in hiding efficiency and brought up a lot of security problems. In particular, an attacker might extract the authentication code and reuse it to log in as a legitimate user when the sender logs off from the system.

Our work focuses on overcoming the aforementioned problems by combining a stronger factor (one-time bio-key) that should be periodically changed, with wavelet-based cryptographic hashing scheme. The result of the combination is one-time image document authentication code *OMAC* with a limited validity period. Thereafter, *OMAC* is embedded within image document as a cover image through reversible data embedding technique [10], which is then transferred via insecure communication channel. The proposed scheme requires no interaction between two entities, and the set of one-time bio-keys is independently generated and updated. It is important to note that the idea of non-interactivity is also used in recently cryptographic scheme [11]. The key used in our scheme based on two strong building blocks, namely, biometrically based on the features extraction of entities' irises (difficult to be lost or stolen) [12], and cryptographically based on strong key-based message authentication code (MAC-SHA-256) [13].

The contributions of the proposed scheme are as follows: First, the proposed scheme addresses all previous weaknesses and presents a new, robust, and end-to-end low complexity one-time image document authentication scheme. This scheme provides autonomous biometric key generation, key selection, and key update. Moreover, the scheme is integrated with wavelet-based cryptographic hashing scheme, which means the scheme is biometrically and cryptographically strong against known attacks, such as replay, forgery, *man-in-the-middle* (MITM), offline guessing, dictionary, *denial-of-service* (DOS) and brute force. Second, this scheme does not attract the attention of eavesdroppers because the one-time image document authentication code *OMAC* is concealed within cover image through wavelet-based steganography. Lastly, a one-time bio-key requires no interaction between the sender and the receiver. Thus, obtaining the key is difficult for attackers, and this key becomes invalid when a user logs off from the system.

This paper is organized as follows: Section II describes the configuration and verification phases of the proposed scheme. Section III presents a security analysis with respect to known attacks. Section IV provides system evaluation. Section V presents the conclusions.

II. Proposed Scheme

Our aim is to present a robust image document authentication scheme that uses five main components, namely, the sender or *S*, the receiver or *R*, cloud service provider or *CSP*, the iris of *S* or IR_s and the iris of *R* or IR_r. The proposed scheme is composed of two phases, namely, configuration and verification. The former is performed only once; both the sender and receiver receive the bio-shared features vector of entities' irises (*Rv*) and bio-shared master key (*Mk*) (Fig. 1). The latter phase is invoked each time a user sends an authenticated image document to another user. A one-time bio-key and wavelet-based cryptographic hash function MAC-SHA-256 *H*(.) [13] are used together to generate *OMAC*, and then embed it into the image document as a cover image by using reversible data embedding [10] (Fig. 2).

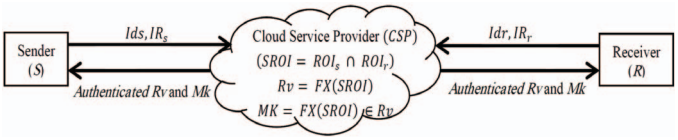

Fig. 1. Configuration phase of the proposed scheme

A. Configuration Phase

In the configuration phase (Fig. 1), the main components *CSP*, *S*, and *R* use *elliptic curve cryptography* (ECC) [14]. These components can run ECC only when biometric data (irises) are transmitted among *CSP*, *S*, and *R* over an insecure channel. This operation is necessary for the configuration phase only and not for the subsequent ones. *CSP* is not required in the verification phase. The following steps are performed during the configuration phase.

- *CSP*, *S*, and *R* run ECC to generate public keys (PU_{CSP}, PU_S, PU_R) and private keys (PR_{CSP}, PR_S, PR_R), respectively. These keys are shared among *CSP*, *S*, and *R* through a secure channel. In particular, these keys are utilized to secure iris transmission from *S* and *R* to *CSP*. Thereafter, both *S* and *R* encrypt their irises IR_s and IR_r, respectively using PU_{CSP} and send them to *CSP*.
- Upon receiving the encrypted IR_s and IR_r, *CSP* decrypts the received irises using the private key PR_{CSP}, saves IR_s and IR_r, and defines regions of interest ROI_s and ROI_r, respectively, as defined in [12]. The preprocessing method described in [12] is used to localize and normalize the ROI_s and ROI_r. Both ROI_s and ROI_r are normalized into a rectangular block of 256×64 pixels. Thereafter, the *CSP* generates a bio-shared *SROI* image by intersecting both ROI_s and ROI_r ($SROI = ROI_s \cap ROI_r$) to compute the vector of features $Rv = Fx(SROI)$ and the shared master key $Mk = Fx(SROI) \in Rv$, as shown in Fig. 1. *FX* refers to a function that extracts features. In details, *FX* employs 2-D Gabor filter [14] and different operations to extract features from the normalized *SROI*.

A set of *2D real Gabor* filters (2GD) with various orientations ($\Theta = 0°$, $45°$, $90°$, and $135°$) are used to filter the normalized bio-shared image *SROI* into four filtered image $SROI_i$, $i = 1, 2, 3, 4$. Each filtered *SROI* image is then equally divided into 32×32 blocks b, and the mean MN of each block is computed. Thus, $32 \times 32 = 1024$ values can be obtained from each filtered *SORI* image. Each value is then normalized to an integer in the range [0,..., 255]. The feature vectors of four filtered images are concatenated 4096-D and multiplied by 8 bits to construct a wide range of 32768-D as follows:

$$Rv = FX(SROI) = 2DG(SROI_{(\Theta=0°,\ 45°,\ 90°\ and\ 135°)}) = SROI^{i}_{(MN(32*32)_b)} | i = 1,\ \ldots,\ 4\ of\ dimension\ 1024 \times 4 = 4096 \times 8 = 32768\ bits$$

Mk is constructed by selecting random positions belonging to Rv. Afterward, *CSP* encrypts Rv and Mk using PU_S and PU_R and transmits them to S and R, respectively. Finally, S and R decrypt the received Rv and Mk using their private keys PR_S and PR_R, respectively. Mk is used in the verification phase for non-interactive one-time biometric key generation, selection, and updating. Rv is used to get a new fresh Mk. An important question is how to generate non-interactive one-time bio-key that facilitates one-time image document authentication code. The verification phase will explain this issue.

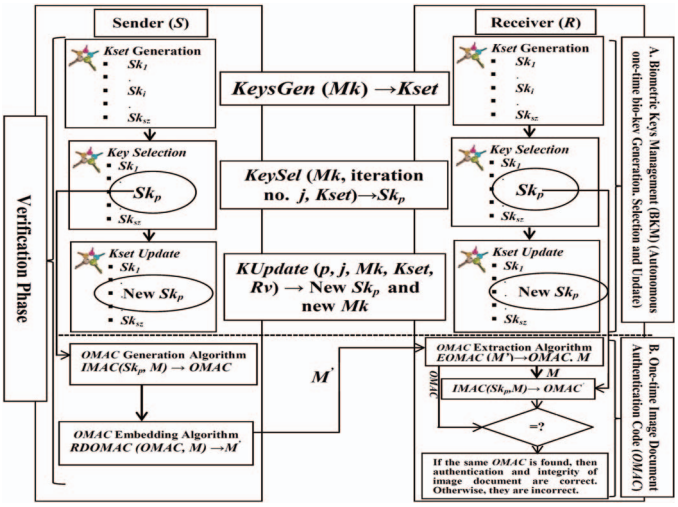

Fig. 2. Verification phase of the proposed scheme. A. Biometric key management *BKM*. B. *OMAC*

B. Verification Phase

After the configuration phase, both *S*/*R* previously shared Mk and Rv through the use of *CSP*, independently run the keys generation algorithm *KeysGen*, through which they generate the set of one-time bio-keys *Kset* ($Sk_i \in Kset$; $i = [1, \ldots, Sz]$; length of Mk) to be used for image document authentication.

When S intends to send an image document M to R or vice versa, it starts the verification phase using selection algorithm *KeySel* to choose the appropriate one-time bio-key to be used by both *S*/*R* (Fig. 2; A. Biometric key management *BKM* block). Subsequently, S runs *OMAC* generation algorithm *IMAC* to integrate the selected one-time bio-key and a wavelet-based hashing scheme for generating *OMAC* and then conceals the *OMAC* within an image document by executing the reversible *OMAC* embedding algorithm *RDOMAC* [10]. The algorithm hides the authentication code and enables the sender to use the image document as a cover image based on reversible data embedding. Also, this algorithm enables the receiver side to extract *OMAC* and restore the embedded bits to source or original bits, which reconstructs the original image document.

Finally, S sends a stego-image document M' to R, which includes *OMAC*. R executes the extraction algorithm *EOMAC* to extract *OMAC* and to recover the original image document M. Then, R compares the extracted *OMAC* with the recomputed one *OMAC'* (Fig. 2; *OMAC* block). After successful verification of the image document authentication and integrity by R, the previously used or the last Mk and the one-time bio-key are removed by both endpoints *S*/*R*. Each time a one-time bio-key and Mk are used or removed, both *S*/*R* must update their one-time key and Mk by replacing the deleted ones with a newly generated ones. The new fresh one-time bio-key and Mk are based on Rv and key update algorithm *Kupdate* (Fig. 2; *BKM* block). Consequently, Rv is necessary for both entities to update autonomously and non-interactively the Mk and *Kset*.

C. Details of The Proposed Scheme

1) Biometric Key Management (Fig. 2; BKM block):

- *Key generation*: After agreeing on a shared Rv and Mk through the configuration phase, the *KeysGen* algorithm is executed autonomously. The algorithm generates a set of one-time bio-keys *Kset*. The length of the master key Mk identifies a given number of one-time bio-keys. For instance, if the length of the master key is 512 (in bits), then *Kset* has 512 one-time bio-keys $Sk_i \in Kset$; $i = [0, \ldots, 511]$. The *Kset* is autonomously generated by both *S*/*R* through the application of different operations on the agreed bio-master key Mk.
  In particular, the *KeysGen* algorithm computes an index $i = [0, \ldots, Sz]$; length of master key. The indices generate different one-time bio-keys *Kset*. For each computed index i, a left i bit(s) rotation is applied to the Mk bits. The result is the message concatenated with index i along with master key Mk as a key to be input by the MAC-SHA-256 function. Thereafter, the result of this function or the 256 bits is one-time bio-key Sk_i added to the *Kset*. This algorithm is iterated based on the length of the master key to generate all the one-time bio-keys *Kset*.
- *Key selection*: When *S*/*R* may wish to send an image document to each other, the *KeySel* algorithm is executed by both *S*/*R* to obtain the same position p of a one-time bio-key Sk_p to generate anonymous image document authentication code. The algorithm enables both endpoints to choose autonomously a one-time bio-key from a set of bio-keys *Kset* generated previously by *KeysGen*. The inputs of *KeySel* are the current iteration number j between S and R ($j = [1, \ldots, Sz]$) as well as the biometric master key Mk along with the pool of keys *Kset*. The *KeySel* returns as its output the position p of the one-time bio-key in the *Kset* ($Sk_p \in Kset$; $p = [1, \ldots, 511]$), which should be used by both *S*/*R*.

In particular, the position p of the bio-key Sk_p to be used by both *S/R* is generated in pseudo-random manner, based on j and *Mk*. That is, the *Mk* is j bit(s) left rotated, then the result is mapped to an integer in the range of [0,..., Sz] by using modulo operation. The mapped value indicates the index p of the one-time bio-key $Sk_p \in Kset;\ p = [0, \ldots,\ Sz]$.

- *Bio-key and master key update*: The *KUpdate* algorithm is executed by both *S/R* to update the *Kset* and master key *Mk*. The algorithm generates a new master key to replace the last used one. Then, using *KeysGen* as well as the new master key to a generate new one-time bio-key and then replace it with old one in the same position p. Consequently, the *KUpdate* maintains the number of keys [0,..., Sz].
  The inputs of *KUpdate* are the position p of the last used one-time bio-key, iteration number j of *KeySel*, last *Mk*, and *Kset* along with *Rv*. The *KUpdate* returns its output as a new one-time bio-key and new *Mk*.
  In particular, the j is concatenated with p, which results in mapping of the integer in the ranges of ([0,..., Sz]; length of *Mk*) and ([0,..., 32768]; length of *Rv*) or j_1 and j_2, respectively. L is obtained by subtracting j_1 from the length of the *Mk*, $L = Sz - j_1$.
  The permutation value Pv is then obtained by subtracting Pv from Rv ($Pv \in Rv(j_2,\ j_2 + L)$), where j_2 is the starting subtracted point, and $j_2 + L$ is the end-subtracted point. Both j_2 and $j_2 + L$ belong to the range of [0,..., 32768]. Thereafter, Pv is substituted with the part of the master key to modulate it. The replaced part starts from the position j_1 with the length L. The result is processed by applying the p left bits rotation to obtain a new *Mk* to replace the old one. Subsequently, the new fresh one-time bio-key is obtained by applying the same operations in *KeysGen* algorithm on new *Mk* (j left bit(s) rotation and MAC-SHA-256) to replace the old one.

2) One-time Image Document Authentication Code (Fig. 2; OMAC block):

- **Sender Side**. When S wants to send an authenticated image document M, the *IMAC* algorithm is executed, beginning with the selection of the appropriate one-time bio-key $Sk_p \in Kset$ using *KeySel* algorithm. Sk_p is used to generate a one-time image authentication code *OMAC*. Subsequently, the *OMAC* is hidden within the image document using *RDOMAC* algorithm and sends to the other endpoint. Fig. 3 represents the proposed diagram for generating *OMAC*. The following steps explain the proposed diagram.

1) Suppose M is the image document dimension $N \times N$ pixels partitioned into equal size and non-overlapping blocks $J \times J$ pixels, where J= 16 [15]. Each block can be represented by BL_k, where $k = [0, \ldots,\ N^2/J^2 - 1]$.
2) Using *KeySel* algorithm to obtain an appropriate one-time key Sk_p; 256 bits long. *Rivest Cipher 4* (RC4) [13] based on permutation key Sk_p is applied for each BL_k. The result is one-time permuted sequence pixels of each BL'_k. RC4 based on one-time bio-key (permutation key) is used for one-time pixel modulation for each block. The modulation is applied before *discrete wavelet transformation* (DWT) is performed.

$$BL'_k = RC4_{Sk_p}(BL_k) \tag{1}$$

3) The second DWT [5] is performed for every modulated blocks $BL'_k;\ k = [0, \ldots,\ N^2/J^2 - 1]$. The coefficients of LL-2 can then be represented by $C(BL'_k)$. Next, quantization is performed for every LL-2 coefficient $Q = q_k(C(BL'_k))$. The final hashed value or *OMAC* is computed using the following equation:

$$OMAC = H(q_0 \| q_1, \ldots, \| q_{(N^2/J^2-1)} \| Sk_p) \tag{2}$$

The use of the one-time bio-key ensures that the scheme has several layers of security, namely, one-time image document authentication code *OMAC*, and one-time randomly modulate image pixels to prevent an attacker from knowing the features space (Eq. 1).

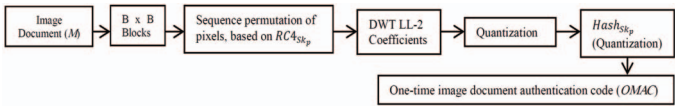

Fig. 3. Block diagram at the sender side for generation *OMAC*

Finally, S sends the stego-image document M', which consists of *OMAC* to R. In particular, the image document M is used as a cover image to embed the *OMAC* previously generated by the *IMAC* algorithm. This embedding can be achieved by using reversible data embedding algorithm *RDOMAC* [10]. *OMAC* is concealed within M proceeds as follows.

1) *OMAC* is 256 bits long; hence, the reversible data embedding technique requires pairs of coefficients to embed a reversible bit. Thus, the image document M is partitioned into equal sizes and non-overlapping blocks of 512 pixels. Consequently, the total number of blocks are $BL_k;\ k = [0, \ldots,\ N^2/512]$.
2) A random block is selected to hide *OMAC*. Sk_p is j bits right rotated, and then the result is mapped to an integer m in the range $[0, \ldots,\ N^2/512]$. m is used to indicate the current block BL_m (512 pixels) to hide *OMAC*; 256 bits long.
3) The second DWT is performed on the selected block BL_m. The coefficients of LL-2 can then be represented by $C_n(BL_m);\ n = [0, \ldots,\ 511]$. For each contiguous pair of coefficients $(C_n,\ C_{n+1}) \in BL_m$, the reversible bit embedding of *OMAC* proceeds as follows:
   - If the values of the pair of coefficients $(C_n,\ C_{n+1})$ are not odd, then transform the $(C_n,\ C_{n+1})$ using the following equation [10]: $C'_n = 2C_n - C_{n+1},\ C'_{n+1} = 2C_{n+1} - C_n$.
     Having done this, set the $LSB(C'_n) = 1$, and then consider the $LSB(C'_{n+1}) = bit \in OMAC$.
   - If the values of the pair of coefficients $(C_n,\ C_{n+1})$ are odd, set the $LSB(C'_n) = 0$, and then consider the $LSB(C'_{n+1}) = bit \in OMAC$.

- **Receiver Side.** Upon receiving an stego-image document M', the receiver runs extracting algorithm *EO-MAC* to extract *OMAC* from a stego-image document M' and to recover the original image document M. The

same processes used in *RDOMAC* are applied to divide the M' into $[0,..., N^2/512]$ blocks and detect the block BL_m (512 pixels) to extract *OMAC*; 256 bits long. The second DWT is performed on the detected block BL_m. The coefficients of LL-2 can then be represented by $C'_n(BL_m);\ n = [0,..., 511]$. For each contiguous pair of coefficients $(C'_n,\ C'_{n+1}) \in BL_m$, the reversible bit extraction technique of *OMAC* proceeds as follows:

- If the $LSB(C'_n) = 1$, then the $LSB(C'_{n+1})$ is extracted and then saved into the detected *OMAC* sequence, set the $LSB(C'_n) = 0$ and $LSB(C'_{n+1}) = 0$. Then, the following equations are applied to obtain the original values of $(C_n,\ C_{n+1})$ [10]:
  $C_n = [2/3\ C'_n\ +\ 1/3\ C'_{n+1}]$,
  $C_{n+1} = [1/3\ C'_n\ +\ 2/3\ C'_{n+1}]$.
- If the $LSB(C'_n) = 0$, then the $LSB(C'_{n+1})$ is extracted and then saved into the detected *OMAC* sequence. The original values of $(C_n,\ C_{n+1})$ are obtained by replacing the LSBs of C'_n and C'_{n+1} with "1" [10].

These above equations are used to extract and restore the embedded bit of *OMAC*. Such restoring operation enables *R* to reconstruct the original image document *M*. Then, *R* based on current iteration number *j* runs the *KeySel* algorithm to select Sk_p. Sk_p and *M* are used together with *IMAC* algorithm. The result is recomputed anonymous one-time image document authentication code $OMAC'$. If $OMAC'$ matches *OMAC*, then *R* ensures the integrity of the image document that *S* has submitted. Otherwise, the verification phase is terminated (Fig 2. *OMAC* block). After each iteration *j*, the *S*/*R* should execute the *Kupdate* algorithm to replace the last used one-time bio-key and *Mk* with a fresh one.

III. Security Analysis

We argue that the proposed scheme has a number of merits and can withstand several threats to security.

Theorem 1. *The proposed scheme can prevent a replay attack.*

Proof: An attacker performs a replay attack by eavesdropping at the login image document, which is transmitted by the rightful *S* to *R*. After the interchange between *S* and *R*, the attacker reuses this image document to impersonate the valid user when he/she logs off the system. In the proposed scheme, the login request of each new sender should be identical to the *CSP* keys (IR_s, IR_r, *SROI*, *Rv*, and *Mk*). Furthermore, each selected one-time bio-key Sk_p from *Kset* is used to generate *OMAC* and is subsequently canceled and replaced with a new bio-key. Therefore, an attacker cannot pass any replayed image document for verification of *R* during the next login. Moreover, computing a new *OMAC* from the last observed one is difficult for the attacker because MAC-SHA-256 and *j*-dependent left rotation are applied on biometric *Mk* for each new one-time bio-key generation. Such operations increase entropy on such Sk_p, thereby increasing entropy for *OMAC* (Eq. 2). Finally, the valid Sk_p that was selected through *KeySel* algorithm is based on pseudo-random manner. Consequently, the attacker fails to perform this attack because our scheme has deflected it. ■

Theorem 2. *The proposed scheme can resist brute force, offline guessing, and dictionary attacks.*

Proof: This attack involves regularly checking all possible keys until the attacker determines the correct one. Assume that an attacker attempts to use cryptanalytic techniques against the crypto-hashed value *OMAC* of *S* . The attacker cannot find the right key Sk_p employed by *S* because the key in our scheme is generated according to the application of different operations on the *Mk*. *Mk* is obtained by extracting (i.e. 512) random positions from bio-shared irises features *Rv* of *R* and *S* ($Rv = Fx(SROI),\ SROI = ROI_s \cap ROI_r,\ Mk \in Rv$) (see Fig.1). In addition, an eavesdropper does not have the main biometric data IR_s, and IR_r generated in the configuration phase to compute the bio-shared feature vector *Rv*. Thus, authenticated *Rv* used by both *S* and *R* is difficult to guess. Biometric features are unique and do not belong to any dictionary [13]. Furthermore, the configuration phase is used only once and then discarded. Thus, expiration time is limited. In addition, the selected Sk_p is used once for image authentication and then deleted and replaced with a fresh one; thus, it is no longer valid. Clearly, this scheme is invulnerable to brute force, offline guessing, and dictionary attacks. ■

Theorem 3. *The proposed scheme is unsuitable to be targeted by DOS attack.*

Proof: This attack generally attempts to prevent or inhibit services of communication facilities and resources. In this case, the authentication system allows a legitimate user to change his or her key or password; hence, the scheme can be targeted by a DOS attack. In our scheme, the one-time bio-key is computed and updated independently by *S* and *R*, and requires no interaction between them. In addition, our scheme does not use a central component in verification phase. However, the proposed scheme uses *CSP* once in the configuration phase and then discards it. Therefore, the proposed scheme is unsuitable to be targeted by DOS attack. ■

Theorem 4. *The proposed scheme can resist MITM and forged attacks.*

Proof: In this attack, the attacker can obstruct the image document between *S* and *R*. The attacker can then use this image document when one user logs out from the *CSP*. In the proposed scheme, the image document is sent from *S* to *R*, or vice versa. *S* generates a one-time bio-key Sk_p to create once sensitive data *OMAC* as a session request to *R*. Thus, sensitive data become useless when *S*/*R* logs out of *CSP*. An attacker who is eavesdropping on the transmission between *S* and *R* will find out that the key Sk_p and *OMAC* are used only once. In addition, the scheme uses the one-time random block (LL-2 coefficients) to hide *OMAC* (256 bits long) within the image document. Hence, computing or disclosing *OMAC* is difficult. Thus, the proposed scheme can resist an MITM attack. ■

IV. System Evaluation

A. Security Comparison

We compare security properties of our proposed scheme with other five authentication schemes in Table I.

B. Performance Investigation

To increase visibility and because of space limitation, 2000 users are used with one 1024×1024 pixels image document

TABLE I. COMPARISON OF AUTHENTICATION SCHEMES

Feature	Proposed scheme	[4]	[6]	[7]	[8]	[9]
C1	Yes	No	Yes	No	No	No
C2	Yes	No	No	No	No	No
C3	Yes	Yes	Yes	No	No	No
C4	Yes	Yes	Yes	Yes	No	Yes
C5	Yes	No	No	No	No	No
C6	Yes	No	No	No	No	No
C7	Yes	No	No	No	No	No
C8	Yes	No	No	No	No	Yes

C1: One-time key; C2: Bio-key; C3: One-time message/image anonymity; C4: Key agreement; C5: Biometrics key management; C6: Secure channel; C7: Cloud environment; C8: Embedding authentication code using steganography

to conceal the authentication code of image document *OMAC* for each user using our scheme. The average total time of the verification phase of each user, including the sender and receiver sides of the scheme, is equal to 0.069s. The average time is obtained after 200 runs of our scheme. Fig. 4 shows that the verification time increases linearly with the number of users. It is clear that our scheme requires not a low but limited overhead in term of computing time of the verification process, since such scheme uses the DWT-based embedding technique to conceal *OMAC* within image document as cover image. However, it is important to point out that the scheme tends to ensure better performance than Shen and Liu [9], in terms of transmission overhead, since image document involved in the verification phase is able to be used in place of the cover image by using *RDOMAC* [10]. In addition, our scheme provides high stego-image PSNR as shown in Table II. Thus, the proposed scheme can strongly combat visual attack.

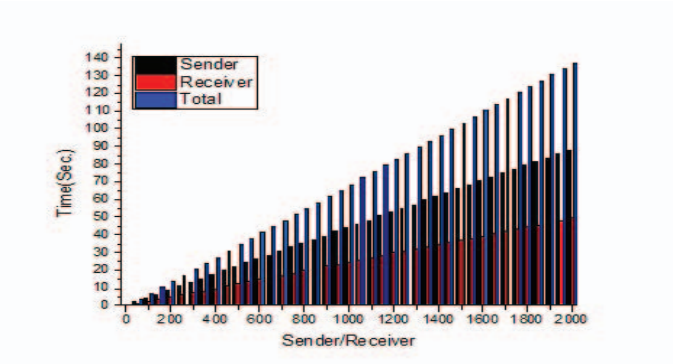

Fig. 4. Average time of the verification phase

TABLE II. MSE AND PNSR OF THE PROPOSED SCHEME

Image document	*OMAC*	Measures	Stego-image document M'
M	ad9bd2fef9c33b7997b752d1bc1d0bc afc8ef46cc30059f3df67bd3ef1d548b	MSE	0.0016 dB
		PSNR	+75.973 dB

V. CONCLUSION

Image document authentication over two endpoints in cloud environment is a challenging task. This paper presents a new and efficient one-time biometric image document code to ensure image document integrity and authenticity between *S* and *R* in a cloud environment. The proposed scheme has several advantages. First, this scheme is robust and efficient in biometric key generation, key selection, and key update procedures as well as in wavelet-based hashing scheme to conduct image document authentication. In particular, both *S* and *R* achieve the key generation, selection and update algorithms independently without any interaction between them. Second, this scheme has been proven by security analysis (Section III) to be biometrically and cryptographically secure. Third, concealing *OMAC* in an image document through reversible data embedding is advantageous because it does not produce additional cover image that becomes extra overhead that should be stored or transmitted along with the document. Overall, the scheme is secure and simple to use.

ACKNOWLEDGMENT

This work is supported by National 973 Fundamental Basic Research Program of China under grant No. 2014CB340600.

REFERENCES

[1] T. Rethika, I. Prathap, R. Anitha, and S. V. Raghavan, "A novel approach to watermark text documents based on Eigen values," *Proceedings of the Ninth International Conference on Network and Service Security (N2S'09)*, France, IEEE, pp. 1-5, 2009.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, John Wiley, vol. 13, no. 18, pp. 1587-1611, Dec. 2013.

[3] A. T. Velte, T. J. Velte, and R. Elsenpeter, *Cloud Computing: A Practical Approach*, McGraw-Hill, 1st Edition, 2010.

[4] N. Rabadi and S. Mahmud, "Drivers' anonymity with a short message length for vehicle-to-vehicle communications network," *Proceedings of the Fifth IEEE Consumer Communications and Networking Conference (CCNC'08)*, Las Vegas, NV, USA, IEEE, pp. 132-133, Jan. 2008.

[5] N. Jamil and A. Aziz, "A Unified Approach to Secure and Robust Hashing Scheme for Image and Video Authentication," *Proceedings of Third IEEE International Congress on Image and Signal Processing (CISP)*, Yantai, China, pp. 274-278, 2010.

[6] Z. Liu, H. S. Lallie, L. Liu, Y. Zhan, and K. Wu, "A hash-based secure interface on plain connection," *Proceedings of the sixth International Conference on Communications and Networking in China (ChinaCom'11)*, Harbin, China, IEEE, pp. 1236-1239, 2011.

[7] S. I. Naqvi and A. Akram, "Pseudo-random key generation for secure HMAC-MD5," *Proceedings of the Third IEEE International Conference on Communication Software and Networks (ICCSN)*, Xi'an, China, pp. 573-577, May, 2011.

[8] C. Chaisri, N. Mettripun, and T. Amornraksa, "Facsimile Authentication Based on MAC," *IT Convergence and Services*, Lecture Notes in Electrical Engineering, Korea, Springer, vol. 107, pp. 613-620, 2012.

[9] J. Shen and K. Liu, "A Novel Approach by Applying Image Authentication Technique on a Digital Document," *Proceedings of International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, Taiwan, IEEE, pp. 119-122, June, 2014.

[10] D. Coltuc and J. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255- 258, 2007.

[11] Y. Yang, "Broadcast encryption based non-interactive key distribution in MANETs," *Journal of Computer and System Sciences*, Elsevier, vol. 80, no. 3, pp. 533-545, 2014.

[12] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical in-dependence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148-1161, 1993.

[13] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson, 6th Edition, 2013.

[14] L. Yu, D. Zhang, and K. Wang, "The relative distance of key point based iris recognition," *Pattern Recognition*, Elsevier, vol. 40, no. 2, pp. 423-430, 2007.

[15] B. Coskun and N. Memon, "Confusion/diffusion capabilities of some robust hash functions," *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, USA, IEEE, pp. 1188-1193, 2006.