Secure Self-Recovery Image Authentication Using Randomly-Sized Blocks

Ammar M. Hassan, Ayoub Al-Hamadi, Bernd Michaelis
IESK, Otto-von-Guericke-University
Magdeburg, Germany
{Ammar, Ayoub.Al-Hamadi}@ovgu.de

Yassin M. Y. Hasan¹, Mohamed A. A. Wahab²
¹EE Dept., Assiut University, ²EE Dept., Minia University
¹Assiut, Egypt, ²Minia, Egypt
¹ymyhasan@aun.edu.eg

Abstract—In this paper, a secure variable-size block-based image authentication technique is proposed that can not only localize the alteration detection but also recover the missing data. An image undergoes recursive arbitrarily-asymmetric binary tree partitioning to obtain randomly-sized blocks spanning the entire image. To enhance reliability of altered block recovery, multiple description coding (MDC) is utilized to generate two block descriptions. Block signature copies and the two block descriptions are embedded into two relatively-distant blocks making a doubly linked chain. The experimental results deposit that the proposed technique successfully both localizes and compensates the alterations. Furthermore, it is robust against the vector quantization (VQ) attack.

Keywords—Authentication; cryptography; multiple description coding; Watermarking;

I. INTRODUCTION

The current advances in information technology and the widespread multimedia applications require efficient methods for guaranteeing privacy, security, protection and integrity of the assorted multimedia data categories. Since many recently developed devices and efficient software products offer consumers worldwide capabilities of flexibly creating, manipulating, and exchanging multimedia data, considerable efforts and contributions have been lately made on digital watermarking that inserts a piece of information (the watermark) into multimedia (host/cover) data for many purposes such as image authentication, copyright protection, fingerprinting and data hiding [1],[2],[3].

Image authentication involves confirmation of the authenticity of the image content, with localized alteration detection, independent of the image format. Various fragile, semi-fragile and self-recovery methods have been introduced for image authentication. The high sensitivity of fragile marks make them attractive for authentication. On the other hand, semi-fragile authentication techniques embed watermarks so robustly to survive (to some, application dependant, extend) various kinds of typical image processing such as lossy compression, filtering. Moreover, to not only localize altered regions but also recover them, self-recovery/embedding authentication techniques have been presented that embed an image approximation into the image itself in a fragile or semi-fragile way using various techniques. This paper mainly addresses fragile block-based self-recovery image authentication purposes. Block-based watermarking techniques equally divide the original image of interest into non-overlapping blocks (may be of 1 x 1 and up to the entire image size) and embed a watermark (for example, the signature of block most significant bits, MSBs, for authentication) into the i-th block (by replacing its least significant bits, LSBs, for example) using a key [1],[2],[3]. An original self-recovery/embedding block-based image authentication technique based on JPEG compression has been introduced in [4]. A JPEG compressed version of each block B_i is inserted into the LSBs of the block B_i + P, where P is a vector of length approximately 1/3 of the image size, with a randomly chosen direction. The algorithm limitations and possible attacks are addressed in [5]. In [6], an algorithm uses a halftone version of the original/host image instead of using a JPEG compression version. The halftone image is pixel-wise permuted using a random generator and embedded into the LSBs of the original image. Fractal codes of a region of interest (ROI), which is chosen as the important object in the image, are used as an approximated version of the ROI [7]. The codes and a watermark are inserted into the LSBs of the host image. Although block-based authentication techniques provide attack localization, less complexity and less memory for intermediate processing, they are vulnerable to the vector quantization (VQ) attack which has been introduced in [8]. Using this attack, one may fabricate an (perceptually indistinguishable) authenticated version of an arbitrary image or deliberately introduce imperceptible changes only to some local original information in an authenticated image. The VQ attack basically works as follows: Assume that an attacker has a set of sufficiently many images authenticated with the same key and knows the block size used. Constructing a VQ codebook using the watermarked blocks, the attacker can search the codebook and find the best match for a given unwatermarked block. Because the watermark embedding and detection are run on independent blocks, the verification process can not sense such VQ forgery/malicious attack.

II. PROPOSED TECHNIQUE

We propose a secure self-recovery image authentication technique that can localize and correct the alteration de-
tects. Furthermore, it successfully thwarts the vector quantization (VQ) attack. To effectively overcome the shortcomings of the existing VQ thwarting techniques [1],[2], we introduce VQ prevention and effective localization schemes, specifically the use of randomly-sized blocks (obtained through random asymmetric binary tree partitioning) and a chain with double links for embedding several signature copies and two descriptions per block in arbitrarily distant blocks. Since the blocks’ sizes are not same, constructing a VQ codebook becomes conceptually infeasible and impractical. Moreover, chaining the multiple embedded signature copies scheme avoids having adjacent blocks hiding signature copies of the same block.

To obtain the randomly-sized blocks, the input image is recursively split into randomly-asymmetric segments until blocks (≥ a minimum size) are obtained. First, the entire image is horizontally/vertically split into two segments based on a random factor. Then, image segments having width/length greater than a specified minimum value(s), alternately (based on a fixed or random binary stream) undergo further vertical/horizontal splitting. The splitting operation is repeated for each divided segment independently and recursively until reaching a set of randomly-sized blocks consuming the whole image.

To increase reliability of missing data recovery, we use multiple description coding (MDC) to generate the block’s descriptions. Actually, MDC methods have been introduced to enhance reliability of error-prone networks. MDC tries hard to deduce the effects of errors using a number of compressed bit streams (descriptions) that can be transmitted through different paths [9],[10]. The descriptions are designed such that each description reconstructs the source with an acceptable quality, and the quality is improved when more descriptions are available to decoder.

The embedding and verification processes are described in next subsections in details.

A. Embedding process

In the embedding process, we insert a block signature copy into the block itself and also insert two block signature copies and two block descriptions into distant blocks. The image of interest is divided into randomly-sized blocks. For each block, the block signature is computed. Then, two block descriptions are generated after the block is resized to be 8 × 8. Depending on the block index, we generate two relatively distant blocks that are used for storing the block signature copies and the block descriptions. Moreover, a block signature copy is inserted into the LSBs of the block itself. For security reasons, the blocks’ signatures and the blocks’ descriptions which will be embedded in the LSBs of a block are encrypted using an encryption algorithm. We can note that the block signature is embedded into two distant blocks and the signature of those distant blocks are embedded into other distant blocks and so on, making a doubly linked chain for all blocks.

Let the original M × N image of interest X be divided into unequal blocks to get Nb blocks as follows:

\[ X = \{ B_1, B_2, ..., B_{N_b} \} \] (1)

For each block \( B_i \), the MSBs of the pixels and the block index are hashed using a hash function such as N-Hash, MD4, MD5, SHA, etc. The output of the hash function \( d \) (digest) may have a long fixed length depending on the specific hash function used. To reduce the digest length, we can choose specific bits as a signature or divide the digest into many equal parts and xor them. So, the vector \( V_s \) that contains the signatures of the blocks is computed as follows:

\[ d_i = H(MSB_{B_i}, \text{index}) \] (2)

\[ S_i = f(d_i) \] (3)

\[ V_s = \{ S_1, S_2, ..., S_{N_b} \} \] (4)

where \( H \) is a hash function, \( f \) is a function to reduce the length of the digest.

To generate two descriptions for each block \( B_i \), \( B_i \) is resized to a 8 × 8 block. Then, the discrete cosine transform (DCT) of the resized block is computed. Chosen low frequency DCT coefficients and the DC are quantized and encoded using a JPEG quantization table and a fixed bit allocation table, respectively. Then, from encoded data, we construct two groups \( D_1 \) and \( D_2 \) where \( D_1 \cup D_2 \) contains all encoded data and \( D_1 \cap D_2 \) contains at least the DC code. \( D_1 \) and \( D_2 \) are two block descriptions. Using this approach, the reconstructed block can be rebuilt with one block description but the reconstruction quality is enhanced with both two block descriptions. To identify indices of two distant blocks for each block \( B_i \), we divide the vector \( V \) that contains the indices of the blocks, i.e., \( V = \{ i, 1, 2, ..., N_b \} \), into five sub-vectors. Each sub-vector is separately interleaved using a seed random generator. Then, the interleaved sub-vectors are concatenated to generate the interleaved vector \( V' \). Therefore, the indices \( j \) and \( k \) of distant blocks are computed as follows:

\[ j = V((i+ | N_b/5 |)mod N_b + 1) \] (5)

\[ k = V((i+ | 3N_b/5 |)mod N_b + 1) \] (6)

The signatures and descriptions of the blocks, which will be inserted into the LSBs of the blocks are stored in the vector \( V_{LSB} \). Each vector element of \( V_{LSB} \) has five fields; first field is for the signature copy of the block itself; the next two fields are for signature copies of distant blocks, and the last two fields are for two blocks’ descriptions. So, \( V_{LSB} \) is
such that:

\[
\begin{align*}
V_{LSB}(i)|field1 &= S_i \\
V_{LSB}(j)|field2 &= S_i \\
V_{LSB}(k)|field3 &= S_i \\
V_{LSB}(j)|field4 &= D1_i \\
V_{LSB}(k)|field5 &= D2_i
\end{align*}
\] (7)

After the \( V_{LSB} \) is completely constructed, each vector element \( V_{LSB}(i) \) is encrypted using a public key encryption algorithm and inserted into the LSBs of the block as follows:

\[
LSBs(B_i) = Encrypt_{Ke}(V_{LSB}(i))
\] (8)

where \( Ke \) is a private key.

B. Verification process

In the verification process, we check the authenticity of a test image and reconstruct its altered regions. The test image \( X \) is divided into blocks as in (1), i.e., \( X = \{B_1, B_2, ..., B_N\} \). For each block, the block signature is computed by (2), (3) and (4) forming the computed signature vector \( V_s \) and the indices of the distant blocks are computed for each block using (5) and (6). Then, we extract the signatures’ copies and the blocks’ descriptions from each block to construct the vector \( V_e \) such that:

\[
V_e(i) = Decrypt_{K_d}(LSBs(B_i))
\] (9)

where \( K_d \) is the public key. The block is considered as an altered block if its computed signature mismatches with the signature that is extracted from the block itself or mismatches with both signature copies, which are extracted from the distant blocks. So, the vector \( V_e \) that contains the authenticity of the blocks is computed as follows:

\[
V_e(i) = \begin{cases} 
1, & V_e(i)|field1 = V_e(i)|field2 \\
\text{or } V_e(i)|field1 = V_e(i)|field3 \\
0, & \text{otherwise}
\end{cases}
\] (10)

where \( j \) and \( k \) are the indices of the distant blocks.

For reconstructing an altered block, we check the status of its distant blocks if at least one of them is not altered, the block description, which has been stored in this distant block is extracted and used for rebuilding the block. Actually, if both distant blocks are not altered, we can extract the two descriptions and rebuild the block with better quality. If both distant blocks are altered, the block is lost forever. In this case, the reconstructed block is remarked as a lost block. Therefore, the reconstructed image \( X_r \) is computed such that:

\[
X_r(i) = \begin{cases} 
dec1(V_e(j)|field4), V_e(i) = 0 \\
\text{and } V_e(j) = 1 \text{ and } V_e(k) = 0 \\
dec2(V_e(k)|field5), V_e(i) = 0 \\
\text{and } V_e(j) = 0 \text{ and } V_e(k) = 1 \\
dec3(V_e(j)|field4, V_e(k)|field5), V_e(i) = 0 \\
\text{and } V_e(j) = 1 \text{ and } V_e(k) = 1 \\
Bl_i, V_e(i) = 0 \text{ and } V_e(j) = 0 \text{ and } V_e(k) = 0 \\
B'_i, V_e(i) = 1
\end{cases}
\] (11)

where \( Bl_i \) is the block that is marked as a lost block; \( dec1, dec2, \) and \( dec3 \) are three block decoding methods.
III. Experimental results

In order to validate the proposed authentication technique, we tested it using several test images. The proposed technique is apparently capable of detecting local alterations attacks and VQ attacks, which replace contiguous blocks by blocks coming from different authenticated images. Thus, we will consider an example to demonstrate the performance in case of a large region in an image coming from another authenticated image. We use RSA as a public key algorithm and SHA (Secure Hash Algorithm) as a hash function. Two LSB plans are exploited for embedding blocks’ descriptions and signatures’ copies. For color images, we apply the proposed algorithm on the three channels of the RGB image. MDC is utilized to generate two descriptions for the original $612 \times 816$ image shown in Fig. 1(a). We can reconstruct an approximated image from each description, as shown in Fig. 1(b) and Fig. 1(c), or from the two descriptions, as shown in Fig. 1(d). The computed signal to noise ratio (PSNR) and the correlation coefficient between the original image and the first approximated images are 25.78 dB and 0.9545, respectively and they are 26.45 dB and 0.9612 for the second approximated image. It is clear that the quality of the reconstructed image is improved when we use both two descriptions where the PSNR and the correlation coefficient are 29.67 dB and 0.9818 respectively. The proposed technique is applied on the original image to yield the authenticated version shown in Fig. 2(a). The PSNR and the correlation coefficient between the original and authenticated images are 49.46dB and 0.9998, respectively. There is no perceptually significant difference between the original and the authenticated images. In Fig. 2(b), an object is removed and precisely replaced by a background coming from another authenticated image without introducing perceptually noticeable difference. Fig. 2(c) shows that the proposed technique effectively detects and localizes the alterations and also it successfully reconstructs the altered regions as shown in Fig. 2(d). The proposed method efficiently recovers the missed object where the PSNR and the correlation coefficient between the original and reconstructed images are 37.51dB and 0.9970, respectively.

IV. Conclusion

A novel self-recovery image authentication technique is proposed. The proposed method is not only capable of localizing the alteration detection but also capable of recovering the missing contents with high reliability using multiple description coding. Furthermore, the proposed technique thwarts VQ attacks using randomly-sized blocks and a doubly linked chain to securely embed the block signature copies into several arbitrary-distant blocks. A public key encryption algorithm is utilized for guaranteeing the security and permitting anyone to test the authenticity of a given image.

ACKNOWLEDGMENT

This work is supported by Forschungsauszeichnung (BMBF-Förderung, FKZ: 03FPB00213), Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by DFG, and BMBF Bernstein-Group (FKZ: 01GQ0702).

REFERENCES


