Counterfeiting Attack on a Lossless Authentication Watermarking Scheme

Yongdong Wu, Changsheng Xu and Feng Bao
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore, 119613
{wydong,xucs,baofeng}@i2r.a-star.edu.sg

Abstract

This paper describes an effective attack on a lossless authentication watermarking scheme [Fridrich et. al 2001]. Given sufficient number of pairs of stego-images and its cover-images, an attacker can obtain the data (e.g., random walk sequence, lookup table) generated from a secret key. With these data, the attacker can easily forge authentic images. The experiment demonstrates this attack is very efficient to the lossless watermarking scheme. To avoid such attack, we modify the process of lookup table generation so that the table is variable with the content of the cover-image.

Keywords: Cover-image attack, lossless authentication watermarking (LAW)

1 Introduction

The rapid development of computer networks and the increased use of multimedia data via the Internet have resulted in the faster and more convenient exchange of digital information. With the ease of editing and perfect reproduction, the protection of ownership and the prevention of unauthorized manipulation of digital audio, image and video materials become important concerns. Digital watermarking, a technique to embed special labels in digital contents, has made considerable progress in recent years. Many image watermarking methods have been developed [Pitas et. al 1996 – Delaigle et. al 1996]. However, most image watermarking methods introduce distortions to the original image content. Although these distortions will not affect the image quality in visual level, it is not acceptable for some applications such as medical imagery, military image analysis and law enforcement. Therefore, it is often desirable to embed a watermark in a reversible way so that the original signal can be recovered from the watermarked signal. Several lossless watermarking schemes [Hongsinger et. al 2001 – Fridrich et. al 2002] are proposed in the last two years. Fridrich proposed a LAW scheme which tries to remove the distortion to recover the original image if the image is authentic. It provides new information assurance tools for integrity protection of sensitive imagery, such as medical images or high-importance military images viewed under non-standard conditions when usual criteria for visibility do not apply. However, an attacker can easily break the method and forge the watermarked images. Figure 1 describes the LAW algorithm for authentication of digital images in the JPEG format [Fridrich et. al 2001]. According to this algorithm, a number of frequency components are selected to authenticate an image. From now on, we define these components as authentication components, and their raster indexes within a block as authentication indexes. In the LAW algorithm, the authors introduced a random walk step. We split it into two steps. One is random walk sequence selection and the other is sequence permutation based on a lookup table generated from a secret key (step 5 in figure 1). Sequentially, a compression step is applied so as to save space for hash value of the original image. A verifier who knows the secret key can reconstruct the random walk sequence and lookup table to recover the authentication coefficients and verify the image.

- Based on the JPEG quality factor, select L authentication indexes $i_1, i_2, \ldots, i_L, j_L$, $0 \le i_l \le 63$, corresponding to middle frequencies.
- Obtain DCT coefficients, $D_k(i)$, $0 \le i \le 63$, $k = 1, \ldots, B$, where B is the total number of blocks in the image. $D_k(i)$ is the i^{th} coefficient of k^{th} block.
- Calculate hash H of the Huffman decompressed stream $D_k(\boldsymbol{P})$ of image $\boldsymbol{P}$.
- Seed a pseudo-random number generator (PRNG) with a secret key and follow a random non-intersecting walk through set $E=\{D_1(i_1), \ldots, D_B(i_1), D_1(i_2), \ldots, D_B(i_2), \ldots, D_1(i_L), \ldots, D_B(i_L)\}$. There are $L \times B$ elements in set E.
- Permute LSBs of the coefficients in set E based on a lookup table generated from the secret key.
- While following the random walk, run the context-free lossless arithmetic compression algorithm for the least significant bits (LSBs) of the coefficients from E. Check for the difference between the length of the compressed bit-stream C and the number of processed coefficients. Once there is enough space to insert the hash H, stop running the algorithm. Denote the set of visited coefficients as E_1, $E_1 \subseteq E$.
- Concatenate the compressed bit-stream C and the hash H and insert the resulting bit-stream into the LSBs of the coefficients from E_1. Huffman compress all DCT coefficients $D_k(i)$ including the modified ones and store the authenticated image on a disk

Figure 1: LAW algorithm [Fridrich et. al 2001]

In the LAW scheme, the random walk sequence and lookup table is uniquely determined by the secret key, and is invariable for all authentication images. This invariable table is vulnerable to cover-image attack. In this attack, we assume the attacker has plenty of cover-images and corresponding stego-images. By comparing a stego-image with its cover-image, the random walk sequence and lookup table, determined by the secret key, are constructed so that the attacker can easily forge authentication images. To countermeasure this attacker, we generate content-related sequence and table from the secret key and the invariable information of the image such as the most significant bits (MSBs).

The paper is organized as follows. Section 2 introduces the cover-image attack and the forgery process. Section 3 presents the countermeasure to the attack. Finally, conclusion and future work are given in section 4.

2 Cover-Image Attack

To launch a cover-Image attack to forge other authenticated images, the attacker should have multiple pairs of original-authenticated images in advance [Fridrich et. al 2001]. This assumption is reasonable to some extent. For example, an attacker can somehow get access to the raw images such as out-of-date images. Additionally, system security should not depend on the confidentiality of an image database. A strong security system should publish all information but the secret keys.

In this attack, the authentication indexes and the lookup table are fixed but unknown to the attacker. The attacker does not know the number of the authentication indexes either. Subsection 2.1 finds the authentication indexes, and subsection 2.2 further reconstructs the lookup table.

2.1 Constructing Authentication Indexes and Random Walk Sequence

In order to minimise the distortion and other artifacts, the authors [Fridrich et. al 2001] selected some authentication components for watermarking based on the JPEG quality factor (step 1 in figure 1). E.g., 3 DCT components 45, 38 and 51 are selected (i.e., L=3) in the paper. An attacker can find the authentication indexes following the process described in figure 2.

In the embedding process mentioned in [Fridrich et. al 2001], only the authentication coefficients may be modified, others are intact. That is, the other values of the stego-image are identical to those of the cover-image. For example, if the values in frequency component 27 of the stego-image are always the same as those of its cover-image, the attacker can make sure that component 27 is not an authentication component. In the algorithm illustrated in figure 2, a non-authentication component is never regarded to be authentication component. The probability that an authentication component is regarded as non-authentication one is only 2^{-B} .

Similarly, if a coefficient is selected for watermarking, the probability of flipping its LSB is 0.5 with one pair of images. Therefore, given n pairs of images, the probability of one coefficient used for watermarking is not found is only 2^{-n}. Therefore, random walk sequence can be reconstructed easily. We focus on reconstructing the lookup table thereafter.

Input: A stego-image $\boldsymbol{Q}$ and its cover-image $\boldsymbol{P}$

Output: Authntication index set $\boldsymbol{\Omega}$

- Initialisation: $\boldsymbol{\Omega}$ = NULL
- As in figure 1, obtain DCT coefficients of cover-image $\boldsymbol{P}$ and stego-image $\boldsymbol{Q}$. One coefficient in image $\boldsymbol{P}$ (or $\boldsymbol{Q}$) is noted as $P_k(j)$ (or $Q_k(j)$ resp.). Where k is the block number, k=1, 2,⋯, B. B is the total number of blocks in the image. j is the index in raster scan order, j=0,1, ⋯, 63.
- For j = 0, 1, ⋯, 63

  for k=1, 2,⋯, B

  if EBP($P_k(j)$) ≠ EBP($Q_k(j)$) then $\boldsymbol{\Omega} = \boldsymbol{\Omega} \cup \{j\}$ and check next index j, where EBP() is the bit plane used to embed the hash value of the cover-image. For example, the LSB plane.

  End

End

Figure 2: Find authentication index

($\boldsymbol{S}_0$, $\boldsymbol{S}_1$) = *FindMapPair*($\boldsymbol{P}$, $\boldsymbol{Q}$)

Input: A cover-image $\boldsymbol{P}$ and its corresponding stego-image $\boldsymbol{Q}$. N is the number of selected coefficients for permutation.

Output: Sets $\boldsymbol{S}_0$ and $\boldsymbol{S}_1$ whose element is a pair of coefficient indexes that may match each other.

- Extract the LSBs $\{P_1, P_2, \cdots, P_N\}$ from the original image $\boldsymbol{P}$, $P_i \in \{0,1\}$ i=1,2, ⋯,N
- Decompress watermarked image $\boldsymbol{Q}$ to get the permutated LSBs $\{Q_1, Q_2, \cdots, Q_N\}$, $Q_i, \in \{0,1\}$ i=1,2, ⋯, N
- Create a set $\boldsymbol{S}_{p0} = \{ i \mid P_i = 0, i=1,2, \cdots, N \}$ and $\boldsymbol{S}_{p1} = \{ i \mid P_I = 1, i=1,2, \cdots, N \}$
- Create a set $\boldsymbol{S}_{q0} = \{ j \mid Q_j = 0, j=1,2, \cdots, N \}$ and $\boldsymbol{S}_{q1} = \{ j \mid Q_j = 1, j=1,2, \cdots, N \}$
- Produce possible set pairs $\boldsymbol{S}_0 = (\boldsymbol{S}_{p0}$, $\boldsymbol{S}_{q0})$, and $\boldsymbol{S}_1 = (\boldsymbol{S}_{p1}$, $\boldsymbol{S}_{q1})$

Figure 3: Find Map Pairs between a stego-image and the corresponding cover-image

2.2 Reconstructing the Lookup Table

The security of the LAW scheme [Fridrich et. al 2001] depends on the secret key, equally, the lookup table generated from the key. If an attack can reconstructs the lookup table, the scheme is totally broken. To this end, the attacker initializes an empty lookup table at first. With

a number of pairs of cover-images and stego-images, the attacker can refine the lookup table step by step. Figure 3 illustrates how to find the potential pairs between one stego-image and its cover-image. Because the LAW scheme permutes LSBs of the DCT coefficients only, an attacker can create two disjoint sets, one is for the authentication coefficients of the cover-image and stego-image whose LSB is 0, and another is for coefficients whose LSB is 1.

Given one pair of stego-image and cover-image, we can add two elements produced in figure 3 into the lookup table. If a second pair of images is applied, we can obtain two more set pairs. Using the second two pairs, the element in lookup table will be refined. Repeat above steps, a complete lookup table will be derived whose element is generated from a PRNG with a secret key as a seed. Figure 4 demonstrates the process.

Input: a number of pairs of cover-images and stego-images

Output: lookup table **T**

Initialisation: **T**←empty

Repeat

- Select one pair of cover-image $\boldsymbol{P}$ and stego-image $\boldsymbol{Q}$ randomly
- Execute *FindMapPair*($\boldsymbol{P}$, $\boldsymbol{Q}$) as figure 3 to get set pairs $\boldsymbol{S}_0 = (\boldsymbol{a}, \boldsymbol{b})$ and $\boldsymbol{S}_1 = (\boldsymbol{c}, \boldsymbol{d})$

  If #**T**= 0, let **T**← { $\boldsymbol{S}_0$, $\boldsymbol{S}_1$ }

  Else for each element $\mathbf{X} = (\boldsymbol{A}, \boldsymbol{B}) \in \mathbf{T}$,

  If $\#\boldsymbol{A} \neq 1$ or $\#\boldsymbol{B} \neq 1$, then compute the common candidates between element **X** and pairs $\boldsymbol{S}_0 = (\boldsymbol{a}, \boldsymbol{b})$, replace element **X** with 2 smaller ones,

  i.e. $\{\mathbf{X}\} \leftarrow \{(\boldsymbol{A} \cap \boldsymbol{a}, \boldsymbol{B} \cap \boldsymbol{b}), (\boldsymbol{A} - \boldsymbol{a}, \boldsymbol{B} - \boldsymbol{b})\}$ and repeat this step with pair $\boldsymbol{S}_1 = (\boldsymbol{c}, \boldsymbol{d})$.

until no replacement occurs

where $\#\boldsymbol{A}$ represents the number of element in set $\boldsymbol{A}$.

Figure 4: Reconstructing the lookup table

2.3 Forgery Images

After obtaining the lookup table, the attacker can easily generate the forgery images. Figure 5 demonstrates the forgery process.

2.4 Analysis and Experiments

Given one pair of images, the number of the elements in lookup table may be quadrupled statistically, or the number of elements in a table element will be reduced to 25% of the previous size roughly. To terminate the table-reconstructing process so that any element in lookup table has only one sub-element, O(*logN*) pairs of images are needed where *N* is the number of coefficients selected in figure 1. Practically, the attack may require extra images because there are invalid set operations.

In the experiment, we test two groups of images, whose size are 256×256 (dotted line in figure 6) and 128×128 (solid line in figure 6). The block size is 8×8. The horizontal axis represents the number of authentication components in a block. The vertical axis represents the number of pairs of stego-images and cover-images in figure 6a and the number of set operations in figure 6b respectively. Figure 6 indicates the attack is effective.

- Obtain the values of quantized DCT coefficients, $D_k(i)$ as figure 1, $0 \le i \le 63$, $k = 1, \ldots, B$, where B is the total number of blocks in the image.
- Calculate hash value H of the Huffman decompressed stream $D_k(i)$ and select the coefficients.
- Select coefficients with the random walk sequence and permute LSBs of the selected coefficients with the constructed lookup table.
- Losslessly Compress the LSBs. Check for the difference between the length of the compressed bit-stream C and the number of processed coefficients. Once there is enough space to insert hash H, stop running the compression algorithm.
- Concatenate bit-stream C and hash H and insert the resulting bit-stream into the LSBs of the coefficients. Huffman compress all DCT coefficients $D_k(i)$ including the modified ones.

Figure 5: forging authentication image

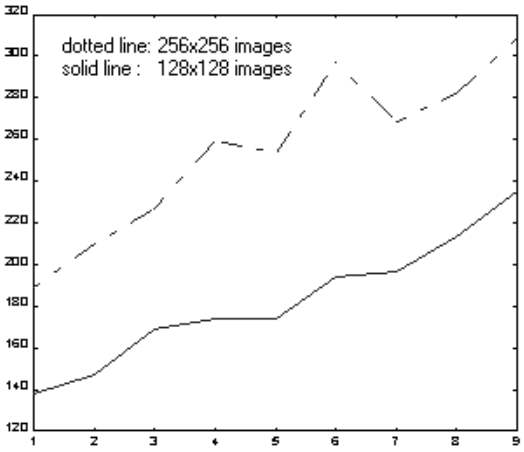
3 Countermeasure

The above attack exploits the weakness that the random walk sequence and lookup table is fixed in the process of embedding images. If we modify them based on the content of the cover-image, the weakness will be removed. Regarding that the 7 MSB bit-planes do not change between the stego-image and its corresponding image, but vary between different images, we can generate a PRNG seed with the secret key and the 7 MSB bit-planes to produce a content-based random walk sequence and lookup table. Since the embedding process is similar to the verifying process, we demonstrate the verifying process which can defeat the cover-image attack in figure 7 only. Those skilled in the art can complete the embedding process easily. The main difference between the revised one and those proposed by Fridrich et al. lies in that the PRNG is generated from the secret key and those LSB bit-planes. Because the sequence and lookup table are different between images, the above attack can not be launched.

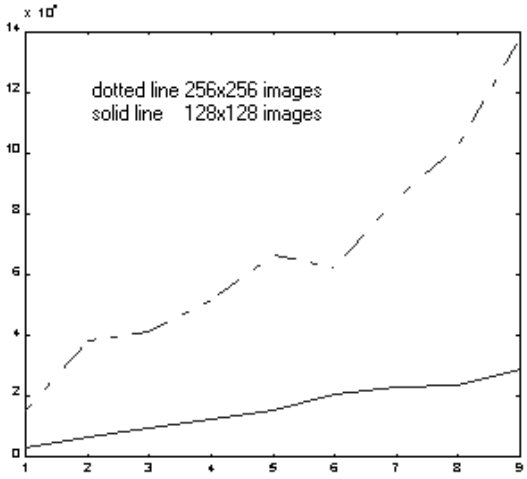
In the updated solution, either content-related random walk sequence or content-related look up table can provide sufficient security performance. Besides the MSB planes, the content-related sequence and lookup table can be generated with the secret key and other recoverable information, for example, the relationship between the DCT coefficients.

4 Conclusion

LAW has an advantage over the scheme [Fridrich et. al 2002) in that LAW applies only one-way function without other cryptography primitives such as cipher algorithm. In this paper, we present a cover-image attack to the LAW scheme [Fridrich et. al 2001] by constructing the random walk sequence and the lookup table generated by a secret key. Although the attacker does not know the secret key, he can easily forge authentication images with them. To foil this attack, we propose a countermeasure so as to produce a content-based random walk sequence or lookup table which is variable to different images.



(a) the number of pairs of images required



(b) the number of set operations required

Figure 6: Attack efficiency

5 References

PITAS, I. (1996): A method for signature casting on digital images, *Proc. IEEE Int. Conf. On Image Processing*, 3:215-218.

WOLFGANG, R.B. and DELP, E.J. (1996): A watermark for digital images, *Proc. IEEE Int. Conf. On Image Processing*, 3:219-222.

COX, I.J., KILIAN, J., LEIGHTON, T. and SHAMOON, T. (1995): Secure spread spectrum watermarking for multimedia, NEC Research Institute, Technique Report 95-10.

SWANSON, M.D., ZHU, B. and TEWFIK, A.H. (1996): Transparent robust image watermarking, *Proc. IEEE Int. Conf. On Image Processing*, 3:211-214, 1996.

DELAIGLE, J.F., VLEESCHOUVER, C.DE and MACQ, B. (1996): Digital Watermarking, *Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques*, 2659: 99-110.

HONSINGER, C.W., JONES, P.W., RABBANI, M. and STOFFEL J.C. (2001): Lossless recovery of an original image containing embedded data, US Patent 6,278,791.

HONSINGER, C.W. (2000): A robust data hiding technique based on convolution with a randomised phase carrier, *Proc. PICS'00*, Portland.

MACQ, B. (2000): Lossless multiresolution transform for image authentication watermarking, *Proc. EUSIPCO*, Finland.

FRIDRICH, J., GOLJAN, M. and DU, R.(2001): Lossless authentication, *Proc. SPIE, Security and Watermarking of Multimedia Contents*, 691-700.

GOLJAN, M., FRIDRICH, J. and DU, R.(2001): Distortion-free data embedding, *4th Information Hiding Workshop*, USA.

FRIDRICH, J., GOLJAN, M. and DU, R. (2001): Lossless Authentication Watermark for JPEG Images, *IEEE International Conference on Information Technology*: *Coding and Computing*, 223-227.

FRIDRICH, J., GOLJAN, M. and DU, R. (2002): Lossless data embedding – New paradigm in digital watermarking, *Special Issue on Emerging Applications of Multimedia Data Hiding*, 2002:185-196.

- Determine the set of L authentication indexes $i_1, i_2, \ldots, i_L$, $0 \le i_l \le 63$.
- Obtain quantized DCT coefficient values, $D_k(i)$, $0 \le i \le 63$, $k = 1, \ldots, B$.
- Seed a PRNG with a secret key and the 7 MSB bit planes. Follow a random non-intersecting walk through set $E=\{D_1(i_1), \ldots, D_B(i_1), D_1(i_2), \ldots, D_B(i_2), \ldots, D_1(i_L), \ldots, D_B(i_L)\}$.
- While following the random walk, run the context-free lossless arithmetic decompression algorithm for the LSBs of the coefficients from E. Once the length of the decompressed bit-stream reaches $B+|H|$ (the number of 8×8 blocks in the image plus the hash length), stop this decompression procedure.
- Recover the LSBs of all coefficients in selected coefficients set with the decompressed bit-stream and the lookup table. Calculate hash H' of the resulting stream of all quantized DCT coefficients $D_k(i)$, $0 \le i \le 63, k = 1,2, \ldots, B$.
- Extract hash H from the decompressed bit-stream. If $H=H'$, the image is authentic and the original image is obtained, otherwise, the image is deemed non-authentic

Figure 7: Algorithm for integrity verification