AN INVERTIBLE WATERMARKING SCHEME FOR AUTHENTICATION OF ELECTRONIC CLINICAL BRAIN ATLAS

Yanjiang Yang, Feng Bao

Laboratory for Information Technology
21 Heng Mui Keng Terrace, Singapore 119613
{yanjiang, baofeng}@lit.org.sg

ABSTRACT

The difficulty in watermarking medical imagery for authentication lies in the fact that watermarking itself should not introduce even one bit of alteration to the images. To this point, the recent invertible watermarking technique can help. However, the existing invertible watermarking schemes are not adaptable to the Electronic Clinical Brain Atlas [1], a kind of "unnatural" palette images with respect to their uncorrelated contents. We develop an invertible watermarking scheme exclusively for authentication of the Electronic Clinical Brain Atlas. What makes our scheme special consists in the candidate points chosen for embedding and the encoding scheme to encode a bit-stream. Furthermore, we present a general framework for invertible authentication watermarking, which encompasses virtually all existing schemes and more importantly, provides higher security over them. As an example, the proposed invertible scheme follows faithfully the framework. Our scheme really solves what others cannot.

1. INTRODUCTION

Due to the special characteristics derived from strict ethics, legislative and diagnostic implications, medical images should be kept intact in any circumstance, which makes medical imagery watermarking a quite unique problem in the sense that authentication itself should not introduce even one bit of alteration to the original content. This makes virtually all existing watermarking schemes inapplicable to medical imagery watermarking since they impose more or less distortions to the original data due to quantization, bit-replacement, and truncation, etc. To this point, fortunately the invertible watermarking technique [2-3,9-10] can help. An invertible watermarking scheme involves inserting a watermark into the host image in a lossless manner, i.e., when the watermark was later extracted, the original image can be recovered completely. This property caters exactly for the watermarking of medical images. In this paper, we explore to authenticate a special type of medical images - Electronic Clinical Brain Atlas [1] by developing an invertible watermarking scheme. These atlases, when working in collaboration with other medical imagery, find a wide spectrum of medical imaging applications [4].

While other type of medical imagery can be invertibly authenticated by the existing invertible watermarking schemes as either "natural" grayscale images or "natural" colour images, the Electronic Clinical Brain Atlas was in an exception as we will see in section 2. An atlas is, in essence, a palette image. However, as demonstrated in Fig. 1., they are not "natural" images as we summarize in the following way their peculiarities.

---**P1**. They are "unnatural" images, as opposed to the "natural" images. By "natural" images, we refer to those we normally see, whose contents are highly correlated.

---**P2**. Each constituting structure in the atlas is of uniform color (homochrome), so that two adjoining structures go along a prominent boundary.

---**P3**. There are always fewer than 256 colors used to label structures, averaging 30-50 colors.

Fig. 1 An Electronic Clinical Brain Atlas (705×820)

2. RELATED WORK

Invertibility of watermarking has been explored for quite a while [5], and the progress within recent two years proves its practical viability in authentication. While the embedding capacity of methods in [9,10], among the few earliest invertible watermarking schemes, was limited, a series of work by Fridrich *et al* [2,3], which cover methods adaptable to virtually all image formats, demonstrate more promising results. In what follows, we choose to briefly review their method for palette images since our scheme is designed for (special) palette images too. Fridrich's method works in two cases:

Case 1: Palette with fewer than 256 colors

For an image whose palette has fewer than 256 distinct colors, choose the most frequently occurring color C in the image, and make some modifications so that C has two entries in the palette, whose indices are i and j, respectively. Then in embedding, scan

the image in a defined pattern. When a pixel with color C is encountered, change its color index to:

i: if "0" to be embedded,

j: if "1" to be embedded.

This scheme works because indices i, j point to the same color. In extraction, the binary message is obtained depending on the pointers i, j to color C when the image is scanned. Embedding in this way does not alter the visual effect of the image at all, so no further reconstruction is needed.

Case 2: Full palette with 256 distinct colors

For an image with full palette having 256 distinct colors, choose two colors C_i (whose color index is i) and C_j (whose index is j) so that

$|C_i - C_j| \leq \Delta$, where Δ is a small value.

Let i denote "0" and j for "1", we get a bit-stream by scanning the image in a defined pattern. Then losslessly compress the bit-stream, and the result is combined with the real payload. In embedding, the combined payload is inserted in such a way that change the index of current C_i or C_j in the scanning path to

i: if "0" to be embedded,

j: if "1" to be embedded.

In extraction, the concatenated payload is taken out depending on the pointer i and j under the same scanning pattern. Then decompress the compressed bit-stream to recover the original indices. It is critical that a color pair with approximate colors exists for the feasibility of the method.

Even though Fridrich's method fits most natural palette images, it is inapplicable to the Electronic Clinical Brain Atlas. First, **P1** and **P2** frustrate case 2. In an atlas, a color pair with approximate colors does not always exist. Even fortunately enough such a color pair is found, since pixels with the same color congregate together (homochrome), thus color flippings are expected to occur in a limited regions. Notwithstanding a minuscule color alteration introduced by each color flipping, the accumulative effect however makes such changes readily detectable. Second, Regarding to **P3**, case 1 can theoretically be utilized, but in practice it cannot be used due to its modification to the palette. First of all, there are a huge number of atlases in the collection, thus modifying all of them is a notoriously demanding task. More importantly, the atlases are strictly protected from modifications, as the same philosophy applied to other medical imagery. In contrast, our invertible watermarking scheme is tailored to circumvent all these limitations.

Prior to our scheme, we present a general framework for invertible watermarking for multimedia authentication.

3. A GENERAL FRAMEWORK

In [6] a watermarking scheme for content authentication is traditionally characterized as

$$m_w = \mathcal{E}_k(m, w) \qquad (1)$$

where $\mathcal{E}_k$ (.), the embedding function, takes as parameters a host image m (we explain the notations in the context of image watermarking) and a watermark (payload) w to output a watermarked image m_w; the embedding function $\mathcal{E}$(.) is usually controlled by a watermarking key k. Its corresponding watermarking detection process is expressed as

$$w = \mathcal{D}_k(m_w) \qquad (2)$$

where $\mathcal{D}_k$(.) is a watermarking extraction function. While (1) and (2) demonstrate to some extent the mechanism of a watermarking scheme, we believe they do not exposit sufficiently the fine particulars in watermarking. So in our general framework for invertible authentication watermarking, we extend the above expressions in a more explicit way, attempting to clarify watermarking as much as possible.

To our understanding, a watermarking scheme is, in essence, a $2\boldsymbol{W}+1\boldsymbol{H}$ problem. In particular, ***W**hat*, ***W**here* and ***H**ow* to embed. Based on this intuition, we define an invertible watermarking model for authentication as

$$\boldsymbol{m_w = \mathcal{E}(m, \mathcal{H}(m', k_1), \mathcal{F}(\tilde{m}, k_2))} \qquad (3)$$

in which we further define

$$w = \mathcal{H}(m', k_1) \qquad (4)$$

$$\ell = \mathcal{F}(\tilde{m}, k_2) \qquad (5)$$

where w is ***W**hat* to embed, ℓ is ***W**here* to embed and $\mathcal{E}$(.) decides ***H**ow* to embed. The corresponding watermarking extraction process is established as

$$\boldsymbol{w = \mathcal{D}(m_w, \mathcal{F}(\tilde{m}, k_2))} \qquad (6)$$

We explain further the framework.

--- In (4), we define the payload w to be jointly determined by m' and k_1, serving the purpose of authentication. $\mathcal{H}$(.) is an one-way function, which can be viewed as a hash function or a digital signature algorithm for ease of understanding. m' is defined as an essential feature of m that succinctly represents m, and k_1 is the authentication key. Note that m' is in most cases m itself, but sometimes comes in other form, such as ROI (Region of Interest). Whatever is, it should satisfy the fact that it is infeasible to change m without changing m'. k_1 comes usually as the secret key shared among involved entities in a symmetric key crytosystem. In short, formula (4) shows that *What* to embed should rely on the image itself and the authentication key.

--- In (5), ℓ, determined by an one-way function $\mathcal{F}$(.), commands the positions where payload w can be inserted. We define $\tilde{m}$ as a unique property of m that survives the embedding process. In other words, given m_w, $\tilde{m}$ can still be readily obtained. In this regard, $\tilde{m}$ can be MSBs in spatial domain or low frequencies in transform frequency domain. The key k_2 serves to conceal the track of embedding such as generating random walks, or the same as k_1 for the authentication purpose too. k_2 may not necessarily be different from k_1, and sometimes even be omitted. By formula (5), we establish that *Where* to embed should depend on the specific image and sometimes a key.

--- In (3), $\mathcal{E}$(.), the embedding function, determines the way m combines w with respect to ℓ. Specially, we define $\mathcal{E}$(.) to have nothing to do with keys. Examples of $\mathcal{E}$(.) include bit-replacement, color flipping, XOR operations, and so on. We further define

$$\mathcal{E}^{-1}(m_w, \tilde{m}, k_2) = \mathcal{E}^{-1}(m_w, \mathcal{D}(m_w, \mathcal{F}(\tilde{m}, k_2))) = m_w',$$

a computationally feasible mapping to recover m_w to m, where m_w' is the reconstructed version of m. To make $\mathcal{E}$(.) invertible, it must satisfy that $\mathcal{H}(m'', k_1) = \mathcal{H}(m', k_1)$, where m'' to m_w' is what m' to m.

--- In (6), since property $\tilde{m}$ is retained in m_w, so that with m_w, $\mathcal{F}(\tilde{m}, k_2)$ is feasible, thus in turn $\mathcal{D}$(.) is feasible.

Compared to the traditional expressions, our framework distinguishes in the following aspects. First, it clarifies in depth the way an authentication watermarking scheme works. Second,

it establishes the role the keys play and further specifies that the embedding function is completely independent of the keys. Last, since w, especially ℓ depend on m, the embedding under our model is content-adaptive in nature. This, in fact, makes attacks to the watermarked images even harder. To see this, let's take the watermarking for JPEG in [2-3] for example: their scheme can be summarized as $m_w = \mathcal{E}(m, \mathcal{H}(m'), \mathcal{F}(k))$, a special case under our framework. Since $\ell = \mathcal{F}(k)$ is non-adaptive, it leaves chances for the adversary to analyze ℓ, in which the security of the scheme totally resides, by mounting some kinds of active attacks. In contrast, our framework leaves no way for such kind of attacks. We thus stress that our framework excels in higher security over others.

The following invertible watermarking scheme for authentication of the atlases follows faithfully this framework.

4. OUR SCHEME

Even though the method in [2-3] fails in our scheme, their way of getting around the invertibility problem of watermarking is conducive to us: embed the information that is used to recover original image together with the payload. We will follow this heuristic. However, the main difficulty in our case is how to find the right places (a path) and how to insert the payload in an atlas, without causing perceptible artifacts.

The same as observed in binary images, the boundaries of homochromous areas may be the only viable places for data hidden. Therefore, the path to insert the payload in an atlas should comprise the boundary points. In the meantime, we should nevertheless be cautious to avoid generating accumulative effect, i.e., revised points should be diffused, better uniformly distributed over the entire atlas. Regarding to how to embed, normal methods such as modifying LSB are inapplicable in our case, because the pixel values of an atlas are color indices rather than RGB values, which means that two pixels with similar values may point to quite different colors. To solve this, we design an encoding scheme to embed a bit-stream into the atlas.

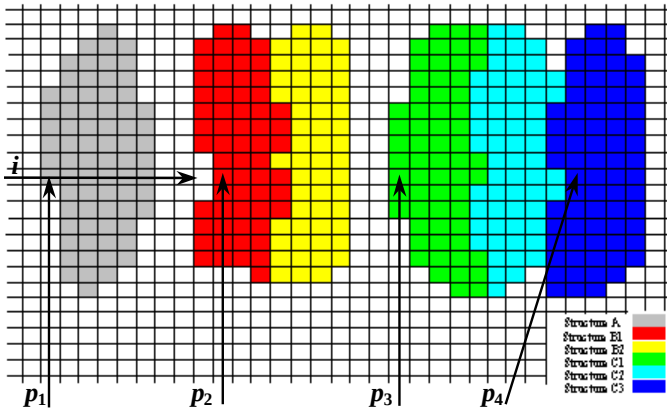

Fig. 2. Illustration of Basic Ideas

We first explain some basic ideas leading to our final scheme and then the complete embedding and authentication algorithms based on these building ideas is presented. We find it is easy to explain our ideas by Fig. 2., which represents virtually all cases of structures bordering each other: structure A stands alone without neighbors; two structures $B1$ and $B2$ border each other; more than two structures in turn border each other (we use $C1$, $C2$ and $C3$ for this case, however generalization beyond this is trivial).

Candidate points selection

Along row i, we in turn pick up p_1 in structure A, p_2 in $B1$, p_3 in $C1$ and p_4 in $C3$ as suitable candidate positions for embedding, and accordingly, A, $B1$, $C1$ and $C3$ are candidate structures, respectively. Note that in each group of neighboring structures, we prefer only the structures with odd sequence number, counting from left to right. The reason is due to our embedding method that otherwise two bordering structures would interfere with each other's embedding.

Encoding scheme

We associate a binary "0" to the even number of points a candidate structure has along row i, and "1" to the odd number of points a candidate structure has. Along row i in Fig. 2, A has 6 points, $B1$ has 4 points, $C1$ has 5 points and $C3$ has 4 points, so the path along i is encoded as "0010".

Embedding method

In order to embed a binary-stream we may need to change the color of the candidate points to their neighboring color. Suppose we want to embed "1001" along row i, then p_1 changes to be the background color, p_2 keeps unchanged, p_3 changes to be background color and p_4 revises its color to be $C2$'s color.

Recovering method

Note that after embedding, the candidate points of a candidate structure may change. In this example, the candidate points of A, $C1$, $C3$ will change after embedding. However, this in no way prevents us from reconstructing the original structures. Let us continue with the above example: suppose for A, which we know its original code is 0, to recover its original form, we simply change the color of p_1 to be A's color. Note that p_1 is determined as it is left next to the new candidate point. Likewise, other candidate structures were recovered.

Based on these ideas, we describe our invertible watermarking scheme below.

Embedding algorithm

1 Scan the atlas m row by row in a fixed pattern, such as from top to bottom. Along each row, pick up the candidate points p_i by **candidate points selection** method, to compose a sequenced set $S = \{p_1, p_2, \ldots\}$.

2 Use the Contour Pick-up Algorithm [7] to obtain the boundaries of all structures in the atlas. Let b denote a combination of the boundaries. Then compute the hash value $H(b, k)$, where k is the authentication key.

3 Seed a PRNG with the concatenation of the authentication key k and the palette of the atlas. The PRNG will generate a random non-intersecting walk through S.

4 Following the random walk, encode the walk by **encoding scheme** as a bit-stream and in the meantime, run an adaptive lossless arithmetic compression algorithm $cmpr(.)$ in a gradual way:

$S_j = \{p_1, p_2, \ldots, p_j\}$ is the bit-stream along the random walk for j steps and $|.|$ denote the length.

while $(|S_i| - |cmpr(S_i)| < |H(b, k)|)$

$i = i + 1$

$S_i = S_i + \{p_i\}$

end

$S' = S_i$

5 Insert $cmpr(S_i)||H(b, k)$ into the path of S' by **embedding method**.

Authentication algorithm

1 Scan the watermarked atlas m_w row by row in the same fixed pattern. Along each row, pick up the candidate points p_i by **candidate points selection**, which in the end compose a sequenced set $S = \{p_1, p_2, \ldots\}$.

2 Seed a PRNG with the concatenation of the authentication key k and the palette of m_w. The PRNG generates a random non-intersecting walk through S.

3 Along the random walk, we encode the walk by **encoding scheme**. We first get H' in the first $|H(b, k)|$ steps. Thereafter, we run the lossless arithmetic decompression algorithm *decmpr*(.) in a gradual way:

$S_j = \{p_1, p_2, \ldots, p_j\}$ is the j-step bit-stream after H' is extracted along the random walk.

while ($|decmpr(S_i)| - |S_i| < |H(.)|$)

$i = i + 1$

$S_i = S_i + \{p_i\}$

end

$S' = S_i$

4 Use $decmpr(S_i)$ to reconstruct the original atlas by **recovering method**, which is denoted as m_w'.

5 Use the Contour Pick-up Algorithm to obtain the combination b' of the boundaries of all structures in m_w'. Then compare the hash value $H(b', k)$ with H', If $H(b', k) = H'$, then the atlas is authentic, otherwise it is tampered with.

Our scheme was strictly in line with the proposed general framework. First, "*What*" to embed is $H(b, k)$. According to the Contour Pick-up Algorithm [7], b changes even one bit of the atlas is altered, thereby b can be viewed as the essential feature of the atlas m, which is m' in the framework. Of course alternatively, we can use the atlas m itself instead of b, but using b will significantly reduce the input to the hash function. Second, "*Where*" to embed is a random walk through a set of possible candidate points, jointly determined by the authentication k and the palette. The palette is the part of the atlas that is retained in the watermarked atlas, hence it is the analogy of $\tilde{m}$ in the framework. As a matter of fact, there are tricks in using the palette this way: on one hand, since each atlas has its distinct palette, the random walk therefore varies from atlas to atlas. This makes our scheme essentially content-adaptive, compared to the non-adaptive methods suggested in [2-3] whose random walk is generated solely by a key. On the other hand, it withstands the attack of swapping entries in the palette, namely, exchange the color values of some entries in the palette while keep the pixel values of the file content intact. We wish the method in [3] would not fall pray to this kind of attack. At last, "*How*" to embed is straightforward, in an invertible manner.

We stress that the random walk generated by the PRNG evenly diffuses the possible accumulative effect introduced by the embedding, which would otherwise be evident if we embed row by row. In the meantime, it reduces the likelihood that an attacker can figure out the embedding path, thereby increasing security.

5. EXPERIMENTAL RESULTS

The experimental results of our method are promising. In experiments, we use an adaptive arithmetic compression algorithm - sixpack program [8] for lossless compression and MD5 for hashing. It is estimated an average of 3500-4000 bits of possible candidate points in an atlas for embedding, thus most of the atlases are watermarkable. We find out only a small number of atlases, each containing so few structures, are not suitable for watermarking, since not enough candidate points are found to accommodate the payload.

6. CONCLUSION

One size cannot fit all. In this paper, we develop an invertible watermarking scheme exclusively for the authentication of the Electronic Clinical Brain Atlas. The specialties of the scheme include the candidate points chosen for embedding and the encoding scheme to encode a bit-stream. Consequently, embedding was achieved along a random walk created by a PRNG among the picked-up candidate points. Such an embedding diffuses effectively possible accumulative degradations in the visual quality of the atlas as well as increases security. The encoding scheme was designed to, along certain rows, associate a binary "0" to the even number of points a candidate structure has and associate "1" to the odd number of points a candidate structure has. Prior to the proposed scheme, we present a general framework for invertible authentication watermarking, attempting to facilitate future development, and our scheme faithfully follows this framework. Although our scheme was intended exclusively for the Electronic Clinical Brain Atlas, they really solve what others cannot.

7. REFERENCES

[1] W. L. Nowinski, *et al*, *Microelectrode-Guided Functinal Neurosurgery Assisted by Electronic Clinical Brain Atlas CD-ROM*, Computer Aided Surgery, pp.115-122, 1998.

[2] J. Fridrich, M. Goljan and R. Du, *Invertible Authentication*, Proc. SPIE Photonics West, vol. 3971, Security and Watermarking of Multimedia Contents III, pp. 197-208, 2001.

[3] J. Fridrich *et al*, *Lossless Data Embedding for All Image Formats*, Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents, pp. 572–583, 2002.

[4] http://www.cerefy.com/

[5] S. Carver, N. D. Memon, B. L. Yeo, M. M. Yeung, *On the Invertibility of Invisible Watermarking Techniques*, ICIP 97, pp. 540-543.

[6] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publisher, 2001.

[7] Y. J. Yang, M. H. Xu, W. L. Nowinski, *A Multilevel Dominant Point Detection for Closed Curved in Atlas-to-Data Registration*. Proc. IEEE-EMBS APBME 2000', pp. 278-279.

[8] G. Held and T. R. Marshall, *Data and Image Compression: Tools and Techniques*, 4th Ed, John Wiley & Sons Ltd, 1998.

[9] B. Macq, *Lossless Multiresolution Transform for Image Authenticating Watermarking*, Proc. of EUSIPCO, 2000.

[10] Honsinger, C. W., Jones, P., Rabbani, M., Stoffel, J. C., *Lossless Recovery of an Original Image Containing Embedded Data*, US Patent application, Docket No: 77102/E–D, 1999