Error-free Authentication Watermarking Based on Prediction-Error-Expansion Reversible Technique

Rongrong Ni†, H. D. Cheng*, Yao Zhao† and Yu Hou†

Institute of Information Science†
Beijing Jiaotong University
Beijing 100044, P.R.China

*Department of Computer Science**
Utah State University
Logan, Utah, U.S.A

E-mail: † rrni@bjtu.edu.cn

* hengda.cheng@usu.edu

Abstract — **Watermarking technology is an efficient method to protect multimedia content. In this paper, an error-free authentication watermarking is proposed based on prediction-error-expansion reversible technique. A binary image is used as an authentication watermark, and embedded in the prediction errors block-wise. A location map is designed and encoded to promise accurate extraction and recovery. A retesting strategy utilizing the parity detection activates the capacity of the ambiguous pixels. In the authentication and recovery period, a watermarked image can be identified as authentic or tampered. If an image is authentic, it can be recovered without errors. The embedded information can be extracted correctly. If an image is a tampered one, the tampered positions can be labeled. The experimental results show the effectiveness and reliability of the proposed method.**

Keywords — **Error-free authentication, reversible watermarking, retesting strategy**

I Introduction

With the development of the Internet and digital technology, multimedia applications are enriching and convenient. However, powerful multimedia processing tools make it possible to alter the content of multimedia. For example, famous PhotoShop of Adobe company is convenient to edit the images. Thus, the integrity and authenticity of image content is threatened.

Authentication watermarking provides a way to protect the image content and identify the tampered parts by embedding an authentication watermark into the original image. Nevertheless, embedding data also changes the content of the image. Even if the image can pass authentication, it is different from the original image. Celik et al. [1] proposed a hierarchical watermarking method using the lowest level for the capability of localization and a high level for resisting the Vector Quantization (VQ) attack [2]. Li et al. proposed a dual-redundant-ring structure which formed a block train and utilized the redundant embedding to improve the security and restoration probabilities [3].

As for some special applications, such as military image transmission and medical image processing, slight changes to the image content cannot be accepted. In these cases, reversible watermarking shows its merit because it can recover the original content without any distortion after data extraction. An effective algorithm is histogram shifting, introduced by Ni et al.[4], which moves the histogram bars to achieve low distortion. Another productive approach is difference expansion algorithm proposed by Tian [5]. The method divides the image into pairs of pixels and uses each legitimate pair for hiding one bit of information. It has high embedding capacity and good quality. Recently, prediction error expansion (PEE) method has been proposed by Thodi et al.[6]. The method uses PEE to embed data, and suggests incorporating expansion embedding with histogram shifting to reduce the location map. Since then, several PEE-based methods have been proposed [7, 8]. In [7], Sachnev et al. proposed a method which combined sorting and two-pass-testing with prediction error expansion method. The algorithm obtains

higher capacity and lower distortion than most of existing reversible watermarking methods. However, these proposed algorithms cannot localize the tampered positions if the image is altered maliciously.

In this paper, an error-free authentication watermarking is proposed based on prediction-error-expansion reversible technique. A binary image is used as an authentication watermark, and embedded in the prediction errors block-wise. A location map is encoded to carry the pixel marks and corresponding block axes. In order to recover the original image during extraction phase, some auxiliary information is required, which is also embedded in the image with the watermark and the location map. Because the actually embedded data is not always identical with the testing bit, some ambiguous pixels appear. A retesting strategy utilizing the parity detection activates the capacity of these ambiguous pixels. In the authentication and recovery period, a watermarked image can be identified as authentic or tampered. If an image is authentic, it can be recovered without errors. The embedded information can be extracted accurately. If an image is a tampered one, the tampered positions can be highlighted. The experimental results show the effectiveness and reliability of the proposed method.

II Primary Ideas

a) *Prediction-Error-Expansion embedding and extraction*

Image pixels are predicted based on the adjacent pixels. Fig.1 illustrates the prediction neighborhood of a pixel, where $x_{i,j}$ is an image pixel, and a, b and c are its neighbors. The predicted value $x^{'}_{i,j}$ is calculated based on Eq.(1)[8].

$$x^{'}_{i,j} = \begin{cases} min(a,b) & \text{if } c \geq max(a,b) \\ max(a,b) & \text{if } c \leq min(a,b) \\ a+b-c & \text{otherwise} \end{cases} \quad (1)$$

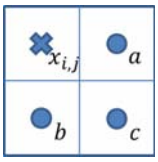


Fig. 1: Prediction neighborhood of a pixel

If the prediction error $d_{i,j} = x_{i,j} - x^{'}_{i,j}$ is within the region $[T_n, T_p]$, $d_{i,j}$ is expanded to $D_{i,j} = 2 \times d_{i,j} + w$. T_n is the negative threshold, T_p is the positive threshold, and w is one watermark bit. Otherwise, if the prediction error belongs to the region $(-\infty, T_n) \cup (T_p, \infty)$, the pixel does not carry any data and the prediction error is simply shifted as: $D_{i,j} = d_{i,j} + T_p + 1$, if $d_{i,j} > T_p$ and $T_p \geq 0$; $D_{i,j} = d_{i,j} + T_n$, if $d_{i,j} < T_n$ and $T_n < 0$. That is[6],

$$D_{i,j} = \begin{cases} 2 \times d_{i,j} + w & \text{if } d_{i,j} \in [T_n, T_p] \\ d_{i,j} + T_p + 1 & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n & \text{if } d_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \quad (2)$$

The watermarked value is computed by $\hat{x}_{i,j} = x^{'}_{i,j} + D_{i,j}$.

During extraction, if $D_{i,j} \in [2T_n, 2T_p + 1]$, $w = D_{i,j} \text{mod} 2$, and $d_{i,j} = \lfloor D_{i,j}/2 \rfloor$; if $D_{i,j} > 2T_p + 1$, $d_{i,j} = D_{i,j} - T_p - 1$; if $D_{i,j} < 2T_n$, $d_{i,j} = D_{i,j} - T_n$. That is,

$$d_{i,j} = \begin{cases} \lfloor D_{i,j}/2 \rfloor & \text{if } D_{i,j} \in [2T_n, 2T_p + 1] \\ D_{i,j} - T_p - 1 & \text{if } D_{i,j} > 2T_p + 1 \\ D_{i,j} - T_n & \text{if } D_{i,j} < 2T_n \end{cases} \quad (3)$$

Then, $x_{i,j} = x^{'}_{i,j} + d_{i,j}$.

b) *Classification and location map construction*

In general, a location map is used to indicate whether a pixel is embedded or not. In early reversible watermarking, the length of the location map equals the length of the image. So, the location map occupies a large amount of embedding space. We propose a new construction method for the location map which needs very limited space.

First of all, every pixel is classified into one of three classes, as suggested by [7]. If a pixel can be modified twice based on Eq.(2), it belongs to Class A; if the pixel is modifiable only once due to overflow or underflow during the second embedding test, the pixel belongs to Class B; if the pixel cannot be modified even once, it belongs to Class C. The two-pass-testing uses bit "1" as an embedding test bit for positive prediction errors, and bit "0" for negative prediction errors. Although the shiftable pixels can be modified, they cannot carry watermark bits. Only can the expandable pixels in Class A and Class B be capable of carrying data. Let the set of expandable pixels in class A be EA. Let the set of expandable pixels in class B be EB.

Only are the pixels in Class B and Class C marked in a location map. One bit is used to remember which class the pixel belongs to, i.e., "0" means Class B and "1" means Class C. Because our authentication method is based on blocks, whose position should be recorded to promise right detection even with tampering. Use L_h bits to record the horizontal axis of the image block, and use L_v bits to record the vertical axis. L_h and L_v depend on both the size of the image and the size of the block (details are given in section III.(a)). For one pixel of Class B or Class C, the total bits in the

location map is $1 + L_h + L_v$. For instance, if 6 bits are used to record the horizontal axis and the vertical axis, 13 bits are needed to record one pixel of Class B (or Class C).

c) Retesting strategy using parity property

In the extraction phase, use once-embedding-test to distinguish Class A, and Class B(or Class C). And further discriminate Class B and Class C using the location map. The shiftable pixels in Class B are shifted in both embedding phase and extraction phase, so this part of pixels can be correctly identified using once-embedding-test during the extraction procedure. As for the expandable pixels in Class B, bit "1" is used to test the overflow for positive prediction errors. When the to-be-embedded bit is "0", some pixels do not exceed 255 even undergoing the second embedding test.As mentioned in [7], during extraction phase, some pixels of Class B will be misclassified to Class A if the actually embedded data equals "0" for positive prediction errors or "1" for negative ones.

To avoid the influence of the misclassification, [7] sacrifices the capacity of Class B. The fixed information is embedded in the expandable pixels in Class B, that is, bit "1" is always embedded in all the positive prediction errors and bit "0" for all the negative prediction errors. However, many images may contain a certain number of pixels of Class B. Rational use of Class B can increase the capacity of the algorithm. We utilize the parity characteristic and retesting strategy to activate the capacity of Class B.

After once-embedding-test during the extraction phase, the pixels are assigned into two parts: Part One contains the pixels without overflow or underflow, and Part Two contains the overflow or underflow pixels. As a result, Part One is the set consisting of Class A and partial Class B, while Part Two is the set containing Class C and part of Class B. It is obvious that the elements of Class B which are attributed in Part One are problem pixels. Since they will cause wrong localization in the location map, the problematic pixels should be identified further.

For the pixels in Part One, a retesting detection is designed to distinguish the ambiguous pixels belonging to Class B. As for the expandable pixels, $\hat{x}_{i,j} = x^{'}_{i,j} + D_{i,j} = x^{'}_{i,j} + 2d_{i,j} + w$. Thus, $\hat{x}_{i,j} - x^{'}_{i,j} = 2d_{i,j} + w$. Due to $2d_{i,j}$ is an even number, $w = LSB(\hat{x}_{i,j} - x^{'}_{i,j})$, here $LSB(\cdot)$ means the Least Significant Bitplane. For the positive prediction errors, if $\hat{x}_{i,j} - x^{'}_{i,j}$ is an even number, the embedded bit is not consistent with the testing bit. Fortunately, the pixel values can be adjusted to fit the case to the two-pass-testing. It is obvious that the difference between embedding "1" and embedding "0" equals 1. Thus, add one to the pixel value $\hat{x}_{i,j}$ and retest the corresponding prediction error using the testing bit "1". For the negative prediction errors, if $\hat{x}_{i,j} - x^{'}_{i,j}$ is an odd number, subtract one from the pixel value $\hat{x}_{i,j}$ and retest the corresponding prediction error using the testing bit "0". If the retesting result shows the pixel is overflow or underflow, it belongs to Part Two. Otherwise, it still belongs to Part One. After the retesting, Part One only contains Class A, and Part Two contains Class B and Class C. Further classification is conducted to distinguish Class B and Class C with the help of the location map. The pseudo-code below describes the retesting process.

Algorithm 1 Retesting Strategy During Extraction

```
Embedding Test to get x̄
if x̄ is overflow or underflow then
  x̂ ∈ Part Two
else
  x̂ ∈ Part One
  For Expanded Pixels in Part One
  if x̂ − x′ is even then
    x̂ ⇐ x̂ + 1
    Retesting using bit "1"
  else
    x̂ ⇐ x̂ − 1
    Retesting using bit "0"
  end if
  if x̄ is overflow or underflow then
    x̂ ∈ Part Two
  end if
end if
```

III Proposed Error-free Authentication based on PEE Reversible Method

Because the prediction value of a pixel is calculated based on the neighbor pixels located on the right or the bottom (Fig.1), the embedding and extraction procedures begin from the point in the lower right corner. A watermarked image can be identified as authentic or tampered. If the image is authentic, it can be recovered without errors. The embedded information can be extracted accurately. If the image is a tampered one, the tampered positions can be highlighted.

a) Data embedding

To recover data, threshold values T_n(5 bits), T_p(5 bits), length of location map (20bits) should be known first, which are embedded into fixed positions in the image by using simple LSB replacement method. The first 30 pixels in the last row are selected as embedding space for these auxiliary information. The original 30 LSBs will be embedded in the image with the watermark. The contents of the location map are also recorded in the

image. Considering prediction structure, we keep the last row and the last column, except for the selected 30 pixels, unchanged during embedding procedure.

The embedding method is designed as follows:

Step 1: Image segmentation and prediction errors calculation. In order to localize the positions of potential tampering, a $M \times N$ original image X is divided into blocks with size 8×8. Thus, there are $L_b = (M/8) \times (N/8)$ image blocks. Authentication watermark W is a binary image, which is also segmented into L_b blocks. For each pixel $x_{i,j}$, compute the prediction value $x^{'}_{i,j}$ and the corresponding prediction error $d_{i,j}$ based on section II.(a).

Step 2: Construct the location map. According to the two-pass-testing, all pixels are classified in one of classes A, B and C. Pixels in Class B and Class C are required to be labeled in the location map. If a pixel belongs to Class B, the corresponding element in the location map is marked as "0"; while if the pixel belongs to Class C, it is marked as "1" in the location map. Meanwhile, determine $L_h = \lceil log_2(M/8) \rceil$, and $L_v = \lceil log_2(N/8) \rceil$. Subsequently, the horizontal and vertical axes of the block that the to-be-labeled pixel belongs to are encoded in the location map. After the creation of the location map L, its length L_n is also obtained.

Step 3: Determine the threshold based on capacity. Shield L_n EA from the beginning pixel of embedding because these EA are used for the location map not for the watermark. Only can the expandable pixels in Class A and Class B be capable of carrying data. Calculate the number of EA and EB in each image block. Adjust the threshold T_p and T_n to make sure that at least one bit of watermark can be embedded in every image block. In details, increase the threshold T_p or decrease T_n, and repeat Step.2 and Step.3.

Step 4: Embed data. The location map L, the authentication watermark W, and the first 30 LSBs are embedded in the image. Because identification of Class B depends on the location map, the location map is only embedded in Class A. Furthermore, the location map is firstly processed in the embedding phase.

In each image block, embedding also begins from the lower right corner to promise the right extraction in the future. Due to the availability of original image during embedding, the prediction values are not affected by the embedded pixels. The bit sequence in watermark block is embedded in the corresponding image block based on Eq.(2). The elements belonging to Class A and Class B are all used to improve the capacity.

As for the 30 LSBs, we arrange them in 30 image blocks to increase embedding burden evenly. As a result, one LSB bit is put before the watermark bits in each chosen block, and stored in the image block with the watermark.

Step 5: Embed auxiliary data. The LSB values of the first 30 pixels at the last row are replaced by the auxiliary data.

After 5 steps, the watermarked image $\hat{X}$ is obtained.

b) Extraction and detection

During extraction phase, the auxiliary data are read firstly, then the location map is extracted to promise the right extraction and recovery. We describe the extraction and detection method in details below,

Step 1: Recover the auxiliary information. Read LSB values from the first 30 pixels in the last row to recover the values of T_n, T_p, and the length of location map.

Step 2: Extract the location map. Except for the last column and the last row, extraction procedure starts from the point at the lower right corner. For each pixel value $\hat{x}_{i,j}$, compute the prediction value based on its prediction neighborhood. Afterwards, the prediction error $D_{i,j}$ is obtained and classified in Part One or Part Two. In detail, bit "1" is used as a testing bit and embedded in the positive prediction errors. While bit "0" is embedded in the negative prediction errors. If the embedded pixel intensity exceeds the pixel range [0,255], it belongs to Part Two. If the pixel value still stays in the pixel range, it belongs to Part One which implies that the pixel may come from Class A. Because some pixels belonging to Class B are misclassified to Class A, further testing is conducted to recognize the problem elements in Part One. For the expandable pixels in Part One, if $\hat{x}_{i,j} - x^{'}_{i,j}$ is even and positive, add one to $\hat{x}_{i,j}$ and retest the corresponding pixel using the testing bit "1". If $\hat{x}_{i,j} - x^{'}_{i,j}$ is odd and negative, subtract one from $\hat{x}_{i,j}$ and retest the corresponding pixel using the testing bit "0". If the retesting result shows the pixel is overflow or underflow, it belongs to Part Two. Otherwise, it still belongs to Part One. After the retesting, Part One is Class A, and Part Two contains Class B and Class C. Extract location map from Class A firstly and recover the pixel values. According to the recorded axes information, the location map matches the corresponding image block.

Step 3: Extract the watermark and detect tampering. Extraction and detection begin from the lower right block. In each block, test every pixel to classify it into Part One or Part Two. Further classification is conducted to distinguish Class B and Class C based on the location map. Then, extract data from Class A and Class B, meanwhile recover the original pixels using the method in section I-I.(a). Compare the extracted watermark with the

original watermark bit by bit. If the compared bits are all same, the image block is authentic. Otherwise, the image block is tampered, and corresponding position of the block is highlighted as white in a labeled image.

Read the first bit of the extracted information from the same 30 selected image blocks as those in the embedding period. These 30 bits are the extracted LSBs.

Step 4: Recover the rest pixels. Replace the first 30 LSB values at the last row with the extracted 30 LSBs.

As a result, the entire watermark is extracted and the original image is restored. If the image is tampered, a labeled image can indicate the tampered positions.

c) Exception discussion

Because the authentication watermark is also evenly divided into small blocks, the length of watermark bits in each block is same. However, the capacity of some image block is not enough to hold corresponding watermark bits. Threshold adjustment strategy promises at least one watermark bit can be embedded successfully. Thus, we will embed watermark bits based on the actual maximum capacity. For example, the length of watermark bits for each block is 16 bits, while the maximum capacity of some corresponding image block is only 5 bits. So, only are the first 5 bits embedded in the image block. During extraction, 5 bits of watermark can be extracted and compared with the first 5 bits read from the original watermark.

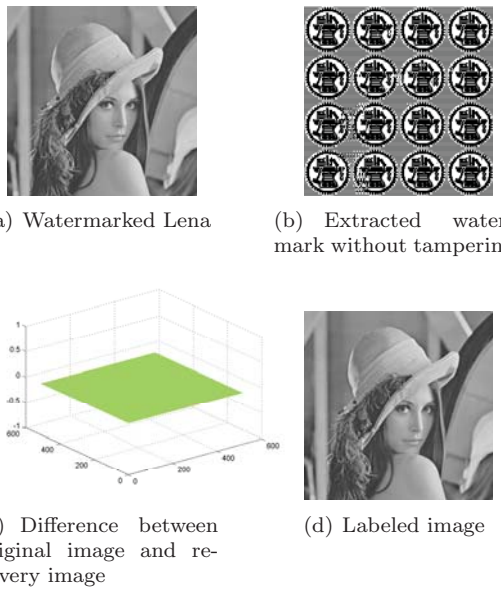
IV Experimental Results

Several 8-bit gray images with size 512×512 and one 256×256 binary watermark are used in the experiments. Fig.2 shows the original images: "Lena", "Plane", and the original watermark.

(a) Lena (b) Plane (c) Binary watermark

Fig. 2: The original images and watermark

The authentication watermark is embedded in "Lena" to get a watermarked "Lena" with PSNR 44.3003dB, as shown in Fig.3(a). Fig.3(b) is the extracted watermark without tampering. It is noticed that some parts are different from the original watermark. This phenomenon illustrates that some image block can only carry the front part of watermark bits, as discussed in section III.(c). All the watermark bits which have been embedded can be extracted accurately. Meanwhile, the image can be restored losslessly, as given in Fig.3(c). The difference between the original image and the recovery image is zero. The labeled image (Fig.3(d)) doesn't highlight any parts.

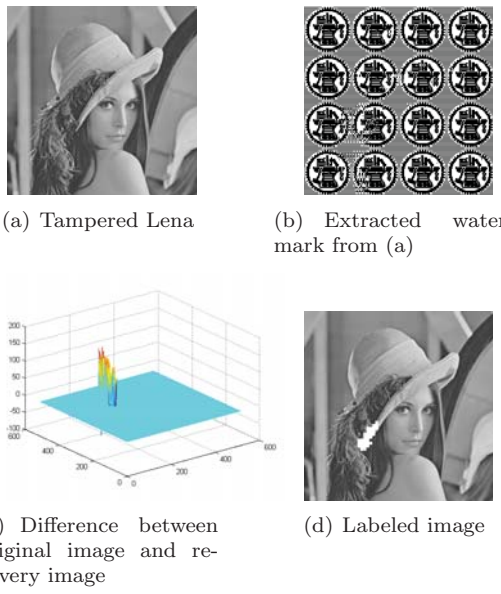


(a) Watermarked Lena (b) Extracted watermark without tampering

(c) Difference between original image and recovery image (d) Labeled image

Fig. 3: Embedding and extraction for Lena without tampering

If tampering occurs, the labeled image can show the detected positions. Fig.4(a) is the tampered Lena, in which some decorations are cut and pasted on her hat. The extracted watermark bits are found wrong at the corresponding positions, as shown in Fig.4(b). Moreover, corresponding recovery of the image is also incorrect, given in Fig.4(c). Fig.4(d) highlights the tampered parts in white.



(a) Tampered Lena (b) Extracted watermark from (a)

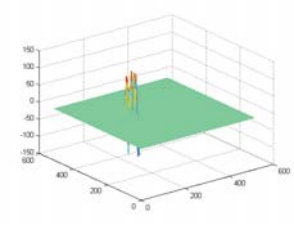
(c) Difference between original image and recovery image (d) Labeled image

Fig. 4: Detection results for Lena with tampering

Fig.5 is another example for the watermarked Plane with PSNR 46.0091dB. Fig.5(a) is the tampered Plane, where the letters are pasted on the body of the plane. The extracted watermark is given in Fig.5(b), and the labeled image indicates the tampered positions(Fig.5(d)). Fig.5(c) shows the difference between the original Plane and the recovery Plane.

(a) Tampered Plane

(b) Extracted watermark from (a)



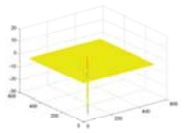
(c) Difference in mesh

(d) Labeled image

Fig. 5: Detection results for Plane with tampering

VQ attack searches a perceptually similar image block from a pool of watermarked blocks, and replaces a block in a test image. Many authentication methods fail to identify the pasted block. While, our proposed algorithm establishes a relationship between blocks in terms of the prediction structure, and detects this kind of replacement. Fig.6(a) is the VQ attacked Lena, in which a block is replaced by another block randomly chosen from other watermarked blocks. It is hard to notice the difference, but Fig.6(c) highlights the tampered parts. The difference between the original Lena and the recovery Lena is shown in mesh in Fig.6(b).

(a) VQ Attacked Lena



(b) Difference in mesh

(c) Labeled image

Fig. 6: Detection results under VQ attack

V Conclusions

In this paper, we propose an error-free authentication watermarking based on prediction-error-expansion reversible technique. A binary authentication watermark is embedded in an image block-wise by using prediction-error-expansion reversible technique. An efficient location map is designed to promise lossless recovery. The proposed method identifies the images as authentic or tampered. The embedded information can be exactly extracted from an authentic image. Meanwhile, the original cover image is restored without distortions. For a tampered image, a labeled image reveals the position of altered parts, and the image is recovered partially.

Acknowledgement

This work was supported in part by 973 Program (2011CB302204), National Natural Science Funds for Distinguished Young Scholar (61025013), National NSF of China (61073159, 61272355), PCSIRT (IRT 201206), Fundamental Research Funds for the Central Universities (2012JBM042).

References

[1] M. Celik, U. Sharma, G. Saber, and A. M. Tekalp. "Hierarchical watermarking for secure image authentication with localization". *IEEE Trans. on Image Processing*, 11(6):585-595, 2002.

[2] M. Holliman, N. Memon. "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes". *IEEE Trans. on Image Processing*, 9(3):432-441, 2000.

[3] C. Li, Y. Wang, B. Ma, and Z. Zhang. "A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure". *Computer and Electrical Engineering*, 37:927-940, 2011.

[4] Z. Ni, Y. Q. Shi, N. Ansari, W. Su. "Reversible data hiding". *IEEE Trans. on Circuits Syst. Video Technol.*, 3: 354-362, 2006.

[5] J. Tian. "Reversible data embedding using a difference expansion". *IEEE Trans. on Circuits Syst. Video Technol.*, 8:890-896, 2003.

[6] D. M. Thodi, J. J. Rodriguez. "Expansion embedding techniques for reversible watermarking". *IEEE Trans. on Image Processing*, 3:721-730, 2007.

[7] V. Sachnev, H. J. Kim, J. Nam, Y. Q. Shi, S. Suresh. "Reversible watermarking algorithm using sorting and prediction". *IEEE Trans. on Circuits Syst. Video Technol.*, 7:989-999, 2009.

[8] D. Coltuc. "Improved Embedding for Prediction-Based Reversible Watermarking ". *IEEE Trans. on Inf. Forensics and Security*, 3:873-882, 2011.