ISBN 978-83-60810-14-9
ISSN 1896-7094

The Quasi-One-Way Function and Its Applications to Image Watermarking

Kazuo Ohzeki
Shibaura Institute of Technology
Toyosu Koutou-ku , Japan
Email: ohzeki @ sic.shibaura-it.ac.jp

Engyoku Gi
Shibaura Institute of Technology
Toyosu Koutou-ku, Japan
Email: 10848@sic.shibaura-it.ac.jp

***Abstract*—This paper describes a one-way function for use in an image watermark to improve authentication ability. The one-way function is popular in cryptography and can prove encryption security. The existence of the one-way function has not been proved yet, while the public key encryption procedure is widely used in practical commerce. The authors proposed a quasi-one-way function, which is weakly defined for practical use. The differences between the strict one-way-function and the proposed quasi-one-way function are discussed. Applications of the quasi-one-way function to image watermarking are shown. Two watermarking systems with SVD method and Jordan canonical form are tried. The robustness of the proposed watermarking method is high.**

I. Introduction

Watermarking is a prospective method to enable forensic tracking of content distribution [1]. A robust one-way operation is expected to be an important component of any system to improve security for authentication of watermarking. The so-called "inversion attack" on watermarked data is based on the addition property of embedding. For an image, embedding is usually done by add operation to cover data. If that addition can be subtracted inversely by an inverse element in that algebraic domain, the inversion attack can be committed easily on any newly determined watermark [2]. The problem originates from this algebraic system's inclusion of an inverse element. A one-way function is a breakwater to prevent inversion attacks and improve system security. Several one-way function candidates have been discussed in the encryption world. Though procedures based on the one-way function are widely used in practical commerce and authentication, the existence of the one-way function has not been proved by a mathematical method.

In this paper, we re consider such a quasi-one-way function and also compare it with the strict one-way function. Watermark embedding is an addition to a cover image. Transforming the cover image into some specific constrained region, the addition is restricted by the region rule. The restriction induces prohibition of subtraction of the data. In such an arrangement, a directional function with addition, whose subtraction can be difficult, is expected to be developed.

A survey paper pointed out the gap between theoretical and practical security. Information-theoretic models for security represent the worst-case, while practical applications exist for optimistic security [3]. Two major security methods are shown; spread-spectrum and asymmetric. This paper takes on a kind of asymmetric method. Four methods were pointed out for security establishment [4]. They are the use of a Trusted Third Party, the asymmetric watermarking scheme, watermark detection using a group of proxies and the Zero-knowledge watermarking detection protocol.

This paper presents a description of a new one-way property for image watermarking. The framework of the algebraic domain is an integer set of image data.

Singular Value Decomposition (SVD) diagonalizes matrix elements by multiplying orthogonal matrices to obtain zero values for off-diagonal elements. The addition of watermarking elements to off-diagonal positions is a quasi-one-way functional operation. The quasi-one-way operation means, in this discussion, that a forward operation result is easily obtained, but it is more difficult to find an inverse value from the result than a forward value. A strict one-way function is not known in mathematical formulation [5]. A quasi-one way operation is one method of realizing a very effective countermeasure against the so-called inversion attack. Many research papers have described inversion attacks, but the best method of dealing with them remains an open problem. Some methods embed a watermark into a random sequence instead of into normal images. One method theoretically embeds a watermark cryptographically. Then it uses a Zero-Knowledge detection method, which differs from an image watermark and is too vulnerable to small attacks.

Gorodetski et al. [5] introduced a watermarking method using SVD in 2001. Ganic [6] surveyed SVD-based watermarking methods in 2003. Many other papers have presented the use of SVD, but the problems of embedding methods remain. Features of SVD that have been discussed include robustness, combination with DCT, and wavelet transformation. A countermeasure to inversion attacks remains elusive. At present, we can not find an SVD watermarking method that features a quasi-one-way function to protect against inversion attacks.

In the study described in this paper, SVD is renovated from a mathematical perspective. Discussions of this matter are revised, and an improved SVD-based watermarking method is developed with a quasi-one-way function to cope

978-83-60810-14-9/08/$25.00 © 2008 IEEE

with inversion attacks. An outline of the SVD watermarking and the quasi-one-way function is proposed in [7]. In this paper, the difference between the strict one-way function and the proposed quasi-one-way function is discussed. Applications of the SVD to image watermarking with one-way directional property are also described. Improved results for a larger size of images compared to the previous experiments [7] are shown, together with experiments using the Jordan canonical form.

II. Image Watermarking

A. Inversion Attack

Embedding a watermark into a cover image usually involves an addition of the image and the watermark. As shown in Fig. 1, for an input image G and a watermark W, we obtain $G_w\ w=G+W$. For this result, an attacker first makes a new his own new watermark W”, which he derives by embedding a watermark information W'' into an image G' to get $G_w{}'=G'+W''$ and subtracting image G' to get $W''=G'w-G'$. Using this watermark W'' , he declares that he had possessed another image G_w-W'' and then embedded watermark W” to get $(G_w-W'')+W''$, which is the same data as G_w . G_w can have the watermark of W'' . Viewing this process, for an arbitrary watermarked image, another person can claim that he had embedded another watermark. This is an inversion attack. The inversion attack always works on the images embedded using a simple addition.

To prevent such inversion attacks it is necessary to introduce a new kind of one-way addition or to transform the embedding procedure into a region with a one-way operation.

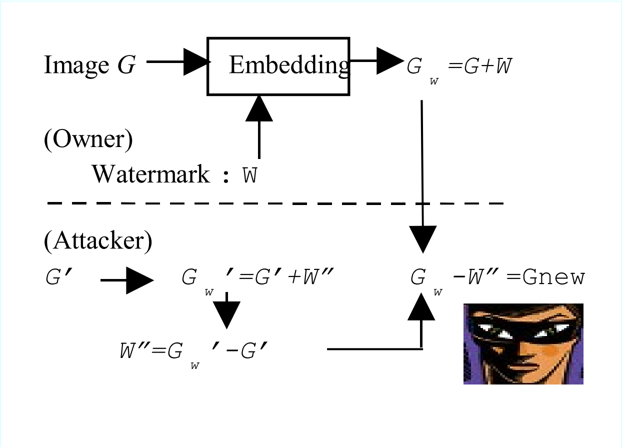

Fig 1. Inversion Attack.

B. Authentication

For authentication of an image watermark, it is necessary to prove that a detected watermark is definitely the embedder’s mark and that the procedure for detecting the watermark from the image is true. If we disclose the procedure to the public, it would mean that everyone knows the detection method and the embedding method and would be able to remove the watermark from that procedure. It is necessary to prove an ownership without disclosing the knowledge of how the watermark truly corresponds to the ownership, because it is also possible to remove the watermark from the image using such knowledge. A zero-knowledge watermark system was presented [5], however, the image is a random number.

A watermark suffers a high error rate as a transmission media, only a restricted number of bits are available for watermarking [8]. At present, it is difficult and impractical for the moment, to construct a watermarking authentication system with public key encryption and zero-knowledge proof. Hence, it would be practical to develop a watermarking system for an application so that watermarks can be embedded into images for web publication. The application would still require authentication at some prescribed level, and to attack such a system would entail certain costs, though it is not perfect to protect the ownership. Examples of authentication methods are to show matrices U and V of singular value decomposition, and to disclose the watermark to someone who can play a role of witness.

III. One-Way Function For Watermarking

A. Necessity For One-Way Function

Let us consider a one-way operation and the inversion attack in regard to the embedding process of a watermark. Given a watermark W and an embedded image G_w $(G_w=G+W)$, whose embedding function is f , the inverse function of the embedding function f is easily obtained as the inverse of the addition. In the case of embedding a watermark in a frequency domain, it is also easy to find the inverse function if the embedding method uses the standard Fourier transform. As long as the embedding process uses an addition for embedding, the inverse function can be easily found using a subtraction. Usually, image data are integer values between [0,255], and all watermarks are represented as additions. It is expected that a one-way operation for the embedding process will be introduced. A one-way function or determining a new algebraic field are candidates for this.

An example of realizing a kind of one-way function is presented [2]. An SVD watermarking system is shown in Fig. 2. An input Image G is decomposed into a diagonal matrix S by the singular value decomposition method using orthogonal matrices U and V . The matrix S is diagonal, and its diagonal elements are singular values while the off-diagonal elements are zeroes. Then, let us consider a procedure in which adding a watermark Ws , which has non-zero elements on off-diagonal positions, recovers image data from $S+Ws$ by multiplying U and V to get G_w . If we re-decompose this embedded image G_w again, we will obtain another set of U' and V' SVD matrices that are different from U and V . So, if we try to subtract W_s from G_w , we cannot get the original image G . Furthermore, elements of the embedded image G_w are usually truncated to integer values, which brings about a non-linear relationship to the watermark embedding system.

B. Quasi-One-Way Function

The o ne-way function is formalized mathematically

A function $f:\{0,1\}^* \rightarrow \{0,1\}^*$ is called one-way if the following two conditions hold [9]:

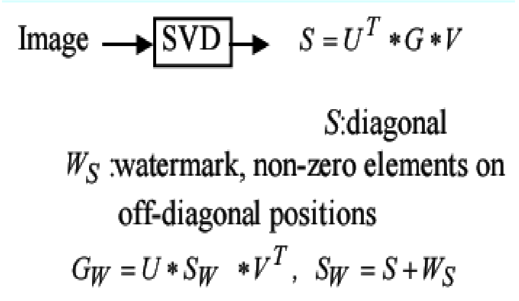
Image
SVD
$S = U^T * G * V$
S:diagonal
W_S :watermark, non-zero elements on off-diagonal positions
$G_W = U * S_W * V^T$, $S_W = S + W_S$

Fig. 2 SVD Watermarking

1. Easy to Evaluate: There exists a probabilistic polynomial-time algorithm *A L* such that *A L* (*x*) = *f* (*x*) for every *x*.
2. Hard to Invert: For every probabilistic polynomial-time Turing machine *M* and for any polynomial *p(n)* , for all *n>N*,

$$\Pr\left(M\left[f\left(U_n\right),1^n\right]\in f^{-1}f\left(U_n\right)\cap\sum{}^{n}\right)<\frac{1}{p(n)} \qquad (1)$$

where, U_n is a uniform probability distribution and the probability is taken over in {0,1}* and the coin tosses in *M* .

Several one-way function candidates have been discussed. It is computationally difficult to obtain an original number from a squared number or from a multiple of two prime numbers under a modulo rule, as indicated by the Robin function. Given a number, there is no fast algorithm of factorization of the number into prime factors. P ro ving the existence of a one-way function is still an open problem. However, difficulty in find ing prime factors is common in practical applications without mathematical verification. This indicates that there should be more sub-classes for practical applications below the polynomial time *(P)* . The term “quasi-one-way function” was mentioned by Whitfield Diffie in a paper [7] which said that “a quasi one-way function is not one-way in that an easily computed inverse exists. However, it is computationally infeasible , even for the designer, to find the easily computed inverse. Therefore a quasi one-way function can be used in place of a one-way function with essentially no loss in security ”. Based on this concept, it is useful to newly create practical computational classes below polynomial. Here, we will extend the statement from computationally infeasible to having more complex mandatory operations than that for forward operation as an inverse function. It is a more realistic and constructive approach to weaken the definition and utilize the number of times of a finite computation.

Definition of a quasi-one-way function:

For a function *y=f(x)* , evaluating the minimum number of times of the forward operation *y=f(x)* and the inverse operation $x=f^{-1}(y)$, a quasi-one-way function should have a larger number of inverse operations than forward operations, as shown by formula (2).

$$\min\left(Num\left(y=f(x)\right)\right)<\mathrm{Min}\left(Num\left(x=f^{-1}(y)\right)\right) \qquad (2)$$

It is recommended that the number of operations of the inverse function is noted.

1. C. SVD and Quasi-ONE-WAY FUNCTION

We will consider the relation between the SVD watermark embedding in Fig. 2 and the quasi-one-way function. If we are given a SVD watermark embedded image G_w and its watermark W_s , then to derive SVD decomposing matrices *U* and *V* may be possible by correcting SVD of G_w using W_s if we neglect the quanti z ing error of G_w . However, the watermark embedded image G_w is truncated to an integer value, which means G_w loses some of the information required to recover the correct SVD decomposing matrices *U* and *V* . It is anticipated that the inverse function will require several times as many operations as the forward function. Concerning the image size of *nXn* , the number of forward operations is order $O(n^3)$ for a non-sparse matrix. The uniqueness of the SVD, which was already proved, governs the inverse operation to be difficult.

IV. Singular Value Decomposition

A. Method 1

In consideration of the preceding section, after embedding a watermark into the singular value matrix *S* , it is not necessary to re- apply SVD. Moreover , a quasi-one - way characteristic of SVD lies in the fact that the off-diagonal position values are zeros. N o positive or negative values exist in the off-diagonal positions . Therefore, to put a value on an off-diagonal position is a quasi-one-way operation. For that reason, adding watermark values at off-diagonal positions is an important quasi-one-way operation. As long as the quasi-one-way operation is effective, it is difficult to find another watermark with an appropriate pair of *U* and *V* of SVD for the embedded image, G_W .

To improve the SVD-based watermarking method, it is merely necessary to remove the operation of the second SVD. The embedded watermark is *SS=S+aW* . Then, the inverse SVD is performed to obtain an embedded image,

$$G_W = U * SS * V^T$$

If another SVD is applied to this G_W , then

$$G_W = U_W * S_W * V_W^T .$$

In general, $SS \neq S_W$, $U \neq U_W$, and $V \neq V_W$.

It is noteworthy that *SS* has off-diagonal elements other than zero, although S_W has only diago nal elements; from S_W, an embedded watermark cannot be detected.

On the other hand, the first and real owner who embedded the watermark can detect *SS* and *W* using *U, V* and *S* . This is “Method 1” of improved SVD-based watermarking with a quasi-one-way function.

However, this Method 1 present s a problem. Alt hough neither *SS* n or *W* can be derived directly from G_W , another effective *U’, V*’ and *W* ’, which m ight differ from the correc t *U, V* and *W* , can be computed using basic linear algebra. Consequently, an inversion attack can be made on this Method 1. Fo r example, after another SVD,

$$G_W = U_W * S_W * V_W^T .$$

Using a proper regular matrix T, $S_W = U_W^T * G_W * V_W$ can be modified, by multiplying T from the left and T^{-1} from the right, to

$$\mathrm{T} * \mathrm{U}_\mathrm{W}^\mathrm{T} * \mathrm{G}_\mathrm{W} * \mathrm{V}_\mathrm{W} * \mathrm{T}^{-1}$$

Putting $T_U = T, T_V = T^{-1}$, we obtain

$$T_U * S_W * T_V = T_U * U_W^T * G_W * V_W * T_V \quad , \quad (3)$$

using a pair of example matrices for T, as

$$T_U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \varepsilon & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_U^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\varepsilon & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_V \quad .$$

Then, (3) is modified to yield the following [10].

$$T_U * S_W * T_V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \varepsilon & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & s_4 \end{bmatrix} * T_V$$

$$= \begin{bmatrix} \mathrm{s}_1 & 0 & 0 & 0 \\ 0 & \mathrm{s}_2 & 0 & 0 \\ 0 & 0 & \mathrm{s}_3 & 0 \\ 0 & 0 & 0 & \mathrm{s}_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \varepsilon\mathrm{s}_1 - \varepsilon\mathrm{s}_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \mathrm{S}_\mathrm{W}^* \qquad (4)$$

Subsequently, (4) is decomposed into a sum of two matrices as,

$$S_W^* = S_W^D + W' \quad ,$$

where S_W^D is a diagonal component and W' is an off-diagonal component.

On the other hand, (3) can be decomposed into a product of two matrices as follows:

Using $T_U * S_W * T_V = T_U * U_W^T * G_W * V_W * T_V$, we obtain

$$(T_U * U_W^T)^{-1} * T_U * S_W * T_V * (V_W * T_V)^{-1} = I_W \quad .$$

Now we can find another SVD for the embedded image G_W with re-defined decomposition matrices as

$$U^* = (T_U * U_W)^{-1} * T_U \quad , \quad V^{*T} = T_V * (V_W * T_V)^{-1} \quad .$$

The following (5) can be inferred from the above [10]. In fact, T_U is a versatile matrix for any watermarked image embedded using Method 1.

$$U^* V^{*T} = (T_U * U_{W^T})^{-1} * T_U * T_V (V_W * T_V)^{-1} \qquad (5)$$
$$= I .$$

B. Method 2

An improved method was proposed to overcome the problem presented in the preceding section [10] . This Method 2 was devised because Method 1 includes the defect that using an appropriate orthogonal matrix T , the diagonal matrix S can be transformed easily into another matrix with off-diagonal components , which breaks the quasi-one-way function. The diagonal matrix S and watermark matrix W are first reviewed to formulate Method 2.

For image matrix G , SVD produces a pair of orthogonal matrices, U and V , where U stands for column transformation and V is the row transformation. Multiplying an orthogonal matrix *"T"* to the diagonal singular matrix S can generate the watermark matrix W, which contains non - zero off-diagonal components. In fact, SS has two expressions,

$$SS = S + W$$

and

$$SS = T_U * S \quad .$$

Then, from $S + W = T_U * S$,

$W = (T_U - I) * S$ or $T_U = (S + W) * S^{-1}$

are obtained. Observing the latter formula, in the case of matrices S and T_U in Method 1, W is not regular, and its diagonal components diminish. The rank of the watermark matrix W increased if the number of embedded watermarks increased . Increasing the rank of W tends to decrease the rank of T_U .

Based on these considerations , Method 2 proposes a non-regular matrix T_U . To reduce the rank of T_U , a partial copy of S into W generates the linearly dependent matrix SS and consequently , $T_U = (S + W) * S^{-1}$ is also non-regular. In SVD, the rank of $U^* = T_U * U_W$ decreases and is not regular. By this operation, the embedded image G_W is decomposed as,

$$G_W = U_W^{**} * S_W^{**} * V_W^{**} \quad .$$

The rank of S_W^{**} , U_W^{**} and V_W^{**} decreases. Thereby it is computationally difficult to obtain regular matrices which match the original G_W from these reduced rank matrices . The simplest example of W is,

$$W(i,i+1) = S(i,i) \ , \ W(i+1,i) = S(i+1,i+1).$$

A flowchart of Method 2 is shown in Fig. 3.

V. Experimental Results

In this section, the proposed SVD-based watermarking algorithm is described using numerical data to confirm the operational methods in detail.

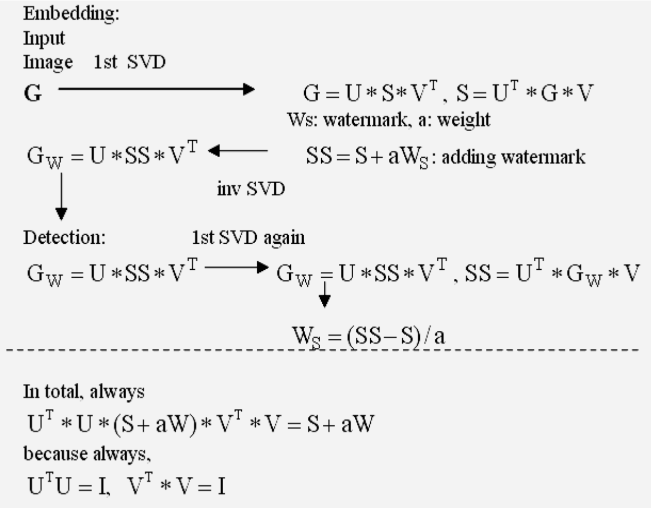
Embedding:
Input
Image 1st SVD
$G \longrightarrow G = U*S*V^T,\ S = U^T*G*V$
Ws: watermark, a: weight
$G_W = U*SS*V^T \longleftarrow SS = S + aW_S$: adding watermark
inv SVD
Detection: 1st SVD again
$G_W = U*SS*V^T \longrightarrow G_W = U*SS*V^T,\ SS = U^T*G_W*V$
$W_S = (SS-S)/a$
In total, always
$U^T*U*(S+aW)*V^T*V = S+aW$
because always,
$U^TU = I,\ V^T*V = I$

Fig 3.: Proposed SVD-Based watermarking embedding and detection.

A. Basic Analysis

For a 4 × 4 image **G** , SVD is shown as the following.

$$\mathbf{G} = \begin{pmatrix} 132 & 122 & 114 & 108 \\ 122 & 116 & 110 & 106 \\ 110 & 116 & 106 & 107 \\ 104 & 107 & 109 & 99 \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} 447.9 & 0 & 0 & 0 \\ 0 & 13.0 & 0 & 0 \\ 0 & 0 & 6.5 & 0 \\ 0 & 0 & 0 & 1.1\ 5 \end{pmatrix}$$

$$\mathbf{U} = \begin{pmatrix} -0.533 & -0.652 & 0.108 & -0.528 \\ -0.507 & -0.252 & 0.003 & 0.824 \\ -0.490 & 0.415 & -0.747 & -0.172 \\ -0.468 & 0.582 & 0.656 & -0.112 \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} -0.524 & -0.827 & 0.114 & 0.168 \\ -0.515 & 0.118 & -0.439 & -0.727 \\ -0.490 & 0.407 & 0.769 & -0.051 \\ -0.469 & 0.370 & -0.450 & 0.664 \end{pmatrix}$$

Next, as a watermark matrix **W** , to make the second column of the singular value matrix conform to the third column,

$$\mathbf{W}(2,3)=\mathbf{S}(2,2),\ \mathbf{W}(3,2)=\mathbf{S}(3,3).$$

are processed. Then **SS**=**S**+**W** is obtainable.

$$\mathbf{W}= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 6.5 & 0 \\ 0 & 13.0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{SS} = \begin{pmatrix} 447.9 & 0 & 0 & 0 \\ 0 & 13.0 & 6.5 & 0 \\ 0 & 13.0 & 6.5 & 0 \\ 0 & 0 & 0 & 1.15 \end{pmatrix}$$

Next, the embedded image $\mathbf{G}_w = \mathbf{U} * \mathbf{SS} * \mathbf{V}^T$ is obtained by inverse SVD using **SS** , **U** , **V**.

$$\mathbf{G}_w= \begin{pmatrix} 130.3 & 124.0 & 111.3 & 110.4 \\ 121.8 & 116.7 & 108.7 & 106.8 \\ 118.3 & 113.7 & 104.1 & 102.2 \\ 97.4 & 106.3 & 115.4 & 100.4 \end{pmatrix}$$

An attacker will attempt to find the singular value matrix **SS** to obtain the embedded watermark matrix **W** . To do so, $\mathbf{G}_w$ is decomposed as,

$$\mathbf{G}_w = \mathbf{Uw}^{**} * \mathbf{Sw}^{**} * \mathbf{Vw}^{**\,T},$$

$$\mathbf{Uw}^{**} = \begin{pmatrix} -0.532 & -0.387 & -0.454 & -0.601 \\ -0.509 & -0.175 & 0.840 & -0.071 \\ -0.490 & -0.232 & -0.278 & 0.793 \\ -0.466 & 0.875 & -0.107 & -0.070 \end{pmatrix}$$

$$\mathbf{Vw}^{**}= \begin{pmatrix} -0.524 & -0.684 & 0.197 & -0.468 \\ -0.516 & -0.099 & -0.747 & 0.407 \\ -0.490 & 0.711 & -0.030 & -0.503 \\ -0.469 & 0.130 & 0.634 & 0.601 \end{pmatrix}$$

$$\mathbf{Sw}^{**}= \begin{pmatrix} 447.4 & 0 & 0 & 0 \\ 0 & 20.5 & 0 & 0 \\ 0 & 0 & 1.40 & 0 \\ 0 & 0 & 0 & 0.19 \end{pmatrix}$$

The rank of $\mathbf{Sw}^{**}$ is theoretically 3, although it seems to be 4 , the fourth value is extremely small , almost zero. Most parts of singular values are sufficiently large, indicating that Method 2 is valid for use in pictures of a general nature.

Many other images were tested with application of SVD. Table 1 shows the maximum and minimum of the singular values of matrix **S** . The maximum values are from **S**(1,1) and the minimum values are from the last non-zero diagonal component. The images in Table 1 are all natural ones ; the rank of **S** is the same value as the image size , except for the circles image . It is an artificial image, not a natural one . Such images usually contain the same value in lines as background parts, and the rank might decrease. For all other images shown in Table 1, SVD operations were all well performed : the ranks of the singular matrix *S* are all the same value as the image size , implying that the proposed method is stable in decomposition for many natural scene images.

B. Jordan Canonical Form

Another basic analysis with the Jordan canonical form was carried out. For an image matrix *G* as,

$$G= \begin{bmatrix} 236 & 10 & 11 & 12 \\ 10 & 177 & 12 & 10 \\ 10 & 12 & 174 & 12 \\ 10 & 12 & 14 & 222 \end{bmatrix}$$

Eigenvalues are well obtained because this matrix is regular

$$\Lambda = P^{-1}*G*P = \begin{bmatrix} 249.3 & 0 & 0 & 0 \\ 0 & 217.3 & 0 & 0 \\ 0 & 0 & 179.2 & 0 \\ 0 & 0 & 0 & 163.3 \end{bmatrix}$$

TABLE 1
RANKS OF IMAGES AND THE MAXIMUM AND MINIMUM VALUES OF SINGULAR VALUES . (GIRL * IS A PARTIAL IMAGE FROM THE CENTER OF AN IMAGE GIRL FROM (126,126) TO (129,129). CIRCLES** IS AN ARTIFICIAL COMPUTER GRAPHIC IMAGE.)

image name	size	rank of S	singular value	
			maximum	minimum
girl_*	4×4	4	447	0.2
car	240×240	240	29761	0.4
girl	256 × 256	256	16511	0.3
couple		256	10003	0.1
lena		256	31956	0.0 1
peppers		256	27203	0.1
circles **		125	28509	40
lena		512	38638	0.0 03

where $P = \begin{bmatrix} 0.805 & 0.603 & 0.163 & -0.023 \\ 0.218 & -0.076 & -0.731 & -0.624 \\ 0.222 & -0.100 & -0.562 & 0.779 \\ 0.505 & -0.788 & 0.351 & -0.054 \end{bmatrix}$

To modify the matrix to have the Jordan canonical form, let the third and fourth components of eigenvalues be the same for having double root and putting a value “1” on the off-diagonal right-hand position of the third eigenvalue. This operation forces the matrix to have the Jordan canonical form that contains a watermark. Let the Jordan’s mark W be

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

then, the embedded matrix is,

$$\Lambda_m w = \begin{bmatrix} 249.3 & 0 & 0 & 0 \\ 0 & 217.3 & 0 & 0 \\ 0 & 0 & 179.2 & 1 \\ 0 & 0 & 0 & 179.2 \end{bmatrix} = \Lambda_m + W \quad .$$

Inverse transformation derives the embedded image matrix as,

$$GwI = R(P * \Lambda_m w * P^{-1}) = \begin{bmatrix} 236 & 10 & 11 & 12 \\ 10 & 184 & 4 & 11 \\ 10 & 5 & 183 & 11 \\ 10 & 12 & 14 & 222 \end{bmatrix}$$

where R (*) represents rounding.

Because the uniqueness of the Jordan canonical form is guaranteed by mathematical theory, the obtained image data matrix is the Jordan matrix.

Multiplying a pair of matrices P and P^{-1} can detect the Jordan’s mark.

$$P^{-1} * G_{w^I} * P = \begin{bmatrix} 249.3 & 0 & -0.1 & 0.3 \\ 0 & 217.3 & 0 & -0.2 \\ -0.2 & 0.1 & 179.6 & 1.41 \\ -0.2 & 0.6 & 0 & 178.8 \end{bmatrix} .$$

To get the Jordan canonical form from the truncated matrix, $G_w{}^I$ could not be done by the change of rounding of integer data. Actually, the result of getting the Jordan canonical form from $G_w{}^I$ is,

$$\Lambda_m wI = \begin{bmatrix} 249.3 & 0 & 0 & 0 \\ 0 & 217.3 & 0 & 0 \\ 0 & 0 & 179.4 & 1 \\ 0 & 0 & 0 & 179.0 \end{bmatrix}$$

which has no double root any more and is not a Jordan matrix, but rather a normal regular matrix .

This Jordan canonical form method is theoretically interesting. However, error sensitivity is large. Therefore, it might not be robust. The example above shows a difference of the embedded mark from 1.00 to 1.41. Embedding trials were done for several image datasets, but the example’s performance and robustness remained unsatisfactory. Eigenvalues can be complex even for real integer data.

C. Embedding Watermarks

Based on the above considerations, watermark-embedding experiments were carried out using Method 2. Fig. 4 shows that the embedding was realized by multiplying a matrix $T_{k,k+1}$. The embedded images were modified using JPEG compression. Detection ratios are shown in Fig. 5 . Table 2 shows their embedded position and singular values. The actual added data are,

$SS(k,k+1)=S(k;1,k+1)$ and $SS(k+1,k)=S(k,k)$.

The detection rule for these experiments is that all singular values are maintained at specified levels. The specified levels are half of the original values. The detection ratios shown in Fig. 5 are normalized by the original values. The dotted line at 50% represents the borderline between detectable and non-detectable. Without JPEG compression, 100% detection is achieved because only a small fractional error exists. Compression ratios of 11-13 are the maximum for detection. In the figure, *SVD_min* means the minimum value of two modified SVD values by JPEG at S (k,k) and S ($k+1,k+1$). These values are maintained as larger than the borderline for JPEG compression ratio of 30. In addition, *WM_min* means the minimum value of the embedded two watermarks modified by JPEG at S ($k,k+1$) and $S(k+1,k$). These values are kept larger than the borderline for JPEG compression ra-

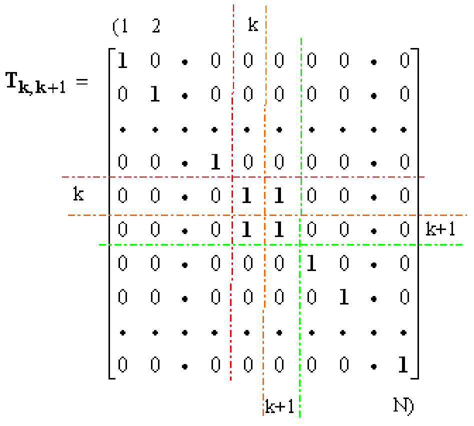

Fig 4. Embedding Matrix $T_{k,k+1}$.

tio 11 or 13. *Ripple _ Max* means the maximum value among all other elements in the neighbouring area except the four elements; *S(k,k)* , *S(k,k+1)* , *S(k+1,k)* and *S(k+1,k+1)* . The *Ripple_Max* area values are originally zeroes and are modified by JPEG compression. The *Ripple_Max* is generally small. For JPEG compression ratio 30, the *Ripple_Max* is less than 15%.

TABLE:2
EMBEDDED POSITIONS AND SINGULAR VALUES.

Image	Embedded Position(1) k	SVD Value	Embedded Position(2) k+1	SVD Value
girl	50	205.06	5 1	195.68
couple	5 0	197.62	51	189.70
lena	85	202.40	86	201.46
peppers	74	204.88	75	199.37
lena2	80	406.16	81	395.78

D. Detection M ethod 2

The detection method above is an elementary version. An improved detection method (DM 2) can be devised from Fig. 5. Room exists between ripples and the 50% detection boundary line. Ripples around diagonal positions with SVD values on them are generally small. Therefore, the detection threshold level *Lw* of embedded watermarks is expected to be smaller than that of a logical value. Furthermore, the SVD value threshold level *Ls* ha s been adjusted. Fig. 6 depicts the results for other images. Detection r atios are improved by this change of threshold. The image quality of JPEG-coded images can be recognized easily for compression ratio s higher than 15 for these experiments. Some degradation is apparent for the compression ratio of 10. The coded images will be shown at oral presentation. Fig. 7 shows a comparison of robustness related to the image size. Images of 256 × 256 were used for the first experiment. An image that wa s twice as large as that used in the previous experiment ha d larger singular values and show ed higher robustness for embedding. Detection ratios of lena_506 we re much higher than the images of 256. For a JPEG compression ratio of 30, the detection wa s well carried out: the larger the image, the higher the obtained robustness.

VI. CONCLUSIONS

Quasi-one-way functions are proposed for watermarking applications. Singular Value Decomposition and Jordan canonical form are introduced for obtaining computationally asymmetrical structures. Jordan canonical form is said to be difficult to calculate precisely for a large matrix. Error sensitivity is large. For real values image data eigenvalues can be complex numbers. Imaginary parts of a pair of conjugate complex eigenvalues resulting from double rooting processing in making Jordan canonical form are likely to be large. Reorganizing the SVD-based watermarking system, an improved SVD-based watermarking method was developed with a quasi-one-way function by reduced rank matrix. The developed matrix with reduced-rank for watermark embedding in singular value decomposition cannot be restored using simple methods, such as simply multiplying a matrix. So-called inversion attacks cannot be activated as long as the proposed quasi-one-way operation framework holds. They would require computationally complex procedures to derive the exact set of singular value decomposition matrices. Results of experiments underscore the effectiveness of the proposed system for smaller sizes of image data. For larger sizes of image data, the detection ratio increases under JPEG compression attacks. In the future, an iterative operation for increasing computational complexity must be investigated. In addition, detection performance can be improved using a smart er and more complicated algorithm

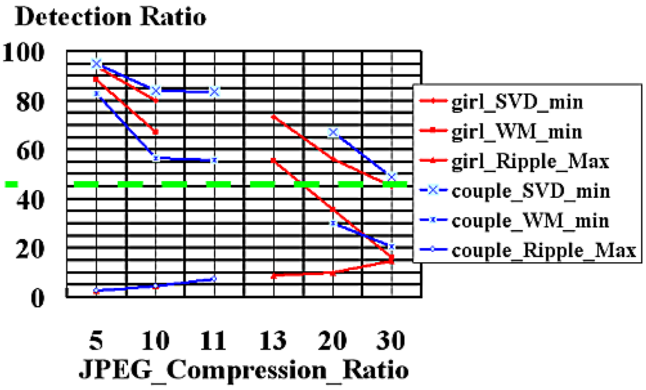

Fig 5. Detection Ratios Depending on Compression

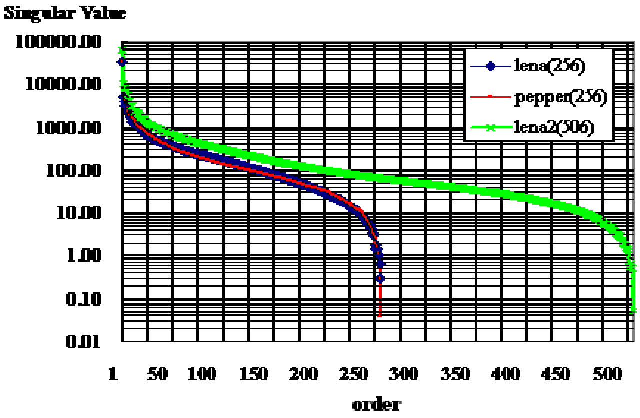

Fig 6. Singular values for larger size of image.

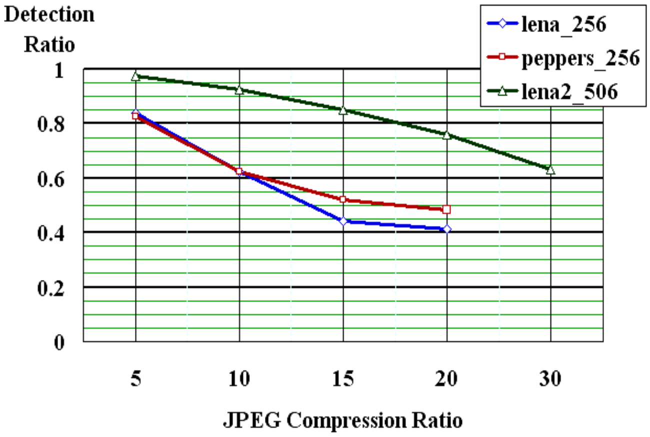

Fig 7 Comparison of detection ratios between large and small sizes of images.

References

[1] Martin Schmucker ed., “First Summary Report on Forensic Tracking”, IST-2002-507932 ECRYPT, D.WVL.7-1.1.pdf , Jan. 2005.

[2] Scott Craver et al., “Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications”, IEEE J. SAC Vol.16,No.4 pp. 573-586, May 1998.

[3] Luis Pérez-Freire et al., “Watermarking security: a survey”. *Transactions on Data Hiding and Multimedia Security I*, 4300:41-72, October 2006.

[4] Qiming Li and EeChien Chang, "ZeroKnowledge Watermark Detection Resistant to Ambiguity Attacks", Proc. ACM Multimedia and Security Workshop pp. 158-163, Sept. 2006.

[5] Goldreich, Oded, “Foundations of Cryptography: A Primer”, Now Publishers Inc 2005.

[6] Ganic, E.et al., “An Optimal Watermarking Scheme Based on Singular Value Decomposition”. Proc. of CNIS. 85-90. 2003.

[7] W. Diffie et al., "New Directions in Cryptography" IEEE Trans.-IT Vol.22, 6, pp.644-654, 1976.

[8] Deepa Kundur, "Authentication Watermarking," ECRYPT *(CMS-2005)*, Sept. 22, 2005.

[9] S. Aida et al., “Average-Time Analysis of One-Way Functions”, IEICE Tech. Rept. COMP99-28, pp.47-54, 1999.

[10] Kazuo Ohzeki and Masaru Sakurai, “SVD-Based Watermark with Quasi-One-Way Operation by Reducing a Singular Value Matrix Rank”, Proc. of e-forensics, Tech B4. WM, 1. Jan., 2008.