Available online at www.sciencedirect.com

SCIENCE DIRECT®

The Journal of Systems and Software 73 (2004) 405–414

The Journal of Systems and Software

www.elsevier.com/locate/jss

Secret image sharing with steganography and authentication

Chang-Chou Lin, Wen-Hsiang Tsai *

Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan, ROC

Received 24 October 2002; received in revised form 30 May 2003; accepted 20 July 2003
Available online 5 December 2003

Abstract

A novel approach to secret image sharing based on a (k, n)-threshold scheme with the additional capabilities of steganography and authentication is proposed. A secret image is first processed into n shares which are then hidden in n user-selected camouflage images. It is suggested to select these camouflage images to contain well-known contents, like famous character images, well-known scene pictures, etc., to increase the steganographic effect for the security protection purpose. Furthermore, an image watermarking technique is employed to embed fragile watermark signals into the camouflage images by the use of parity-bit checking, thus providing the capability of authenticating the fidelity of each processed camouflage image, called a stego-image. During the secret image recovery process, each stego-image brought by a participant is first verified for its fidelity by checking the consistency of the parity conditions found in the image pixels. This helps to prevent the participant from incidental or intentional provision of a false or tampered stego-image. The recovery process is stopped if any abnormal stego-image is found. Otherwise, the secret image is recovered from k or more authenticated stego-images. Some effective techniques for handling large images as well as for enhancing security protection are employed, including pixelwise processing of the secret image in secret sharing, use of parts of camouflage images as share components, adoption of prime-number modular arithmetic, truncation of large image pixel values, randomization of parity check policies, etc. Consequently, the proposed scheme as a whole offers a high secure and effective mechanism for secret image sharing that is not found in existing secret image sharing methods. Good experimental results proving the feasibility of the proposed approach are also included.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Secret image sharing; Steganography; Authentication; Camouflage image; Data hiding; Stego-image; Fragile watermarking; Least significant bit replacement

1. Introduction

Due to fast growth of Internet applications, digitized data becomes more and more popular. Because of the ease of digital duplication and tampering, data security becomes an important issue nowadays. In certain application cases, it is a risk if a set of secret data is held by only one person without extra copies because the secret data set may be lost incidentally or modified intentionally. In some other cases, it might be necessary for a group of persons to share a certain set of secret data. Shamir (1979) proposed first the concept of (k, n)-threshold secret sharing to solve this problem. The scheme is designed to encode a secret data set into n shares and distribute them to n participants, where any k or more of the shares can be collected to recover the secret data, but any $k - 1$ or fewer of them will gain no information about it. After the scheme was proposed, many related topics have been studied (Sun and Shieh, 1994; Chang and Lee, 1993). However, the resulting methods are suitable for only a few types of digital data, such as text files, passwords, encryption/decryption keys, etc.

Because of the drastic expansion of network bandwidths, data flow on networks nowadays include extensively all types of multimedia, including image, audio, video, etc. In particular, how to share a secret image has attracted wide attention in recent years because of the popular uses of images in network applications. Naor and Shamir (1995) proposed first the idea of visual cryptography. The scheme provides an easy and fast decryption process that consists of xeroxing the

* Corresponding author. Tel.: +886-3-5728368; fax: +886-3-5734935.
E-mail addresses: whtsai@cis.nctu.edu.tw, gis85529@cis.nctu.edu.tw (W.-H. Tsai).

0164-1212/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/S0164-1212(03)00239-5

shares onto transparencies and then stacking them to reveal the shared image for visual inspection. The scheme, which differs from traditional secret sharing, does not need complicated cryptographic mechanisms and computations. Instead, it can be decoded directly by the human visual system. However, the scheme is suitable for binary images only and the generated noisy share may be suspicious to invaders. Although an extended scheme for gray-level and color images (Verheul and van Tilborg, 1997; Blundo et al., 2000; Lin and Tsai, 2003) was proposed later, the problem of arousing suspicion still exists.

Steganography is a kind of data hiding technique that provides another way of security protection for digital image data. Unlike utilizing a particular cipher algorithm to protect secret data from illicit access, the purpose of steganography is to embed secret data in preselected meaningful images, called *camouflage images*, without creating visually perceptible changes to keep an invader unaware of the existence of the secret. Numerous schemes have been developed to achieve the goal of data hiding (Bender et al., 1996; Hsu and Wu, 1999; Wu and Tsai, 1998, 1999; Kundur and Hatzinakos, 1999; Adelson, 1990). Differing from previously-mentioned visual cryptography schemes, which generate noisy images as shares that might be suspicious to invaders, the idea of generating shares with meaningful contents is proposed in this study. This enhances the security protection effect. It requires the use of data hiding techniques in the secret sharing process.

On the other hand, it will be advantageous to check in advance the fidelity of all shares before they are used to reconstruct the secret image. This prevents a secret sharing participant from incidental or intentional provision of false share data, causing unsuccessful secret recovery. One way to include such an *authentication capacity* in the secret sharing scheme is to use fragile watermarks. A *fragile watermark* (Lin and Delp, 1999) is a kind of signal, which is designed to be embedded in an image and can be easily destroyed if the watermarked image is manipulated in the slightest manner. By inspecting the existence of the embedded signal in an inspected image, the aim of authentication can be achieved. In this study, a technique of fragile image watermarking is adopted for image authentication during the secret sharing process.

The remainder of this paper is organized as follows. In Section 2, use of the Shamir method for secret sharing is first described. In Section 3, the principle of the proposed approach to secret image sharing with the capabilities of steganography and authentication is described. In Section 4, a detailed algorithm to implement the proposed approach is given. In Section 5, an algorithm for secret recovery is described. Some experimental results are shown in Section 6. Finally, some conclusions and discussions are given in the last section.

2. Use of the Shamir method for secret sharing

The proposed approach to secret image sharing is based on the (k, n)-threshold secret sharing method proposed by Shamir (1979). In this section we describe how to use the Shamir method for conventional secret sharing before describing our approach in the next section.

By the Shamir method, to generate n shares for a group of n secret sharing participants from a secret integer value y for the threshold k, we can use the following $(k-1)$-degree polynomial

$$F(x) = y + m_1 \times x + m_2 \times x^2 + \cdots + m_{k-1} \times x^{k-1} \tag{1}$$

in the following way.

1. Select the number k is to be no larger than n.
2. Choose the $k-1$ integer values $m_1, m_2, \ldots, m_{k-1}$ randomly.
3. Choose freely for the ith secret sharing participant a value of x (denoted as x_i), but all x_i must be distinct from one another.
4. For each chosen x_i, compute a corresponding value of $F(x_i)$ by Eq. (1).
5. Take each pair of $(x_i, F(x_i))$ as a *secret share* and deliver it to a participant.

In the above secret sharing process, the $k-1$ chosen values of m_i need not be kept after all secret shares are generated; they can be recovered, together with the secret value y, from the n secret shares in the secret recovery process as described in the following.

1. Collect at least k secret shares from the n ones to form a system of equations as follows:
$$\begin{aligned} F(x_1) &= y + m_1 \times x_1 + m_2 \times x_1^2 + \cdots + m_{k-1} \times x_1^{k-1}, \\ F(x_2) &= y + m_1 \times x_2 + m_2 \times x_2^2 + \cdots + m_{k-1} \times x_2^{k-1}, \\ &\vdots \\ F(x_k) &= y + m_1 \times x_k + m_2 \times x_k^2 + \cdots + m_{k-1} \times x_k^{k-1}. \end{aligned} \tag{2}$$
Note that the x_i and $F(x_i)$ in (2) above with $1 \leqslant i \leqslant k$ are $2k$ known values collected from the k secret shares.
2. Use a polynomial interpolation technique like the Lagrange method to solve the k unknowns, $m_1, m_2, \ldots, m_{k-1}$, and y, in the k equations in (2) and reconstruct the $(k-1)$-degree polynomial $F(x)$ described by Eq. (1) to be:
$$\begin{aligned} F(x) = {} & F(x_1)\frac{(x-x_2)(x-x_3)\cdots(x-x_k)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} \\ & + F(x_2)\frac{(x-x_1)(x-x_3)\cdots(x-x_k)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} \\ & + \cdots + F(x_k)\frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}. \end{aligned} \tag{3}$$

3. Compute the solution for the secret value y as $F(0)$ which may be derived from (3) above to be

$$y = (-1)^{k-1}\left[F(x_1)\frac{x_2x_3\cdots x_k}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_k)} + F(x_2)\frac{x_1x_3\cdots x_k}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_k)} + \cdots + F(x_k)\frac{x_1x_2\cdots x_{k-1}}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})}\right]. \quad (4)$$

Note that according to Shamir (1979), if fewer than k secret shares are collected, the k unknowns cannot be solved and the desired y value cannot be reconstructed.

3. Principle of proposed approach

The proposed approach provides, in addition to secret image sharing, the capabilities of steganography and authentication, the former being useful for the purpose of hiding the shares and the latter for verifying their fidelity before secret reconstruction. In this section, we sketch the principle of the proposed approach by a simple algorithm, followed by detailed discussions on some proposed techniques for adapting the Shamir method for secret image sharing.

3.1. Proposed adaptation of the Shamir method for secret image sharing

The above Shamir method is useful for sharing secret integer data. To apply the Shamir method to share a secret image, one way is to consider the entire image data as an integer created by concatenating the data bytes of all the image pixels. However, this is impractical because the resulting integer might become too large to be handled in the equations described by (1)–(4). For example, even a small gray-scale image with a very small size of 16×16 will result in an integer possibly of the enormous value of $2^{8\times16\times16}$! In this study, the entire secret image is not taken as a single secret value; instead, each individual pixel is handled as a separate secret integer value, thus avoiding the above-mentioned problem of enormity of the secret data value.

The basic idea of the proposed approach is to compute secret shares from *each* pixel value in a given secret image, and hide them together with certain watermark signals behind a set of corresponding blocks in the camouflage images. As a simple illustration, let S be a single image pixel taken as the secret image, and $B_1, B_2, \ldots, B_n$ be n distinct gray-scale camouflage image blocks, each of the size of 2×2 or 4 pixels. The pixel S or anyone in B_i has a single byte of data to specify its gray intensity value. Let the value of S be denoted as s, which is just the secret data to be shared by n participants. Also, let the four pixels in each B_i be denoted as X_i, W_i, V_i and U_i, and their values as x_i, w_i, v_i, and u_i, respectively. Furthermore, let the eight data bits of X_i be denoted as $x_{i1}, x_{i2}, \ldots, x_{i8}$ and those of W_i as $w_{i1}, w_{i2}, \ldots, w_{i8}$ and so on. Finally, let $b_1, b_2, \ldots, b_n$ be n data bits to be used as watermark signals, with b_i to be embedded in B_i, respectively. An illustration of the locations and the data of the pixels X_i, W_i, V_i, and U_i in each B_i is shown in Fig. 1. The following is a basic algorithm that reveals the principle of the proposed approach to secret image sharing. It is just a sketch of a more detailed algorithm which will be described later in this section.

X_i $x_i = x_{i1}x_{i2}\ldots x_{i8}$	W_i $w_i = w_{i1}w_{i2}\ldots w_{i8}$
V_i $v_i = v_{i1}v_{i2}\ldots v_{i8}$	U_i $u_i = u_{i1}u_{i2}\ldots u_{i8}$

Fig. 1. An illustration of the locations of the pixels in image block B_i and their values.

Algorithm 1. Basic process for secret image sharing with a single image pixel as the secret.

Input : (1) a secret image pixel S with value s; (2) n distinct 2×2 camouflage image blocks $B_1, B_2, \ldots, B_n$; and (3) n watermark signal bits $b_1, b_2, \ldots, b_n$.

Output : n manipulated camouflage image blocks $B'_1, B'_2, \ldots, B'_n$, with n secret shares and watermark signal bits embedded in the blocks.

Steps.

Step 1. Take the value x_i of the top-leftmost pixel X_i of each camouflage image block B_i as the value x specified in Eq. (1).

Step 2. Take the value s of the secret image pixel S as the value y specified in Eq. (1).

Step 3. Choose arbitrarily a set of $k-1$ integer values for use as the m_i in Eq. (1) where $k \leqslant n$.

Step 4. For each x_i, compute the corresponding value of $F(x_i)$ by Eq. (1) to form a secret share $(x_i, F(x_i))$ for each participant in the secret sharing group.

Step 5. Hide the eight data bits of $F(x_i)$ in the data bits of the three pixels W_i, U_i, and V_i of the corresponding camouflage image block B_i.

Step 6. Embed the watermark signal bit b_i also in the data bits of one of the three pixels mentioned in the last step.

Each manipulated camouflage image block B_i' will be called a *stego-image block* in the sequel, with the meaning that it is expected to have secret hiding or steganographic effect.

3.2. Merit of proposed secret image sharing approach

The above algorithm illustrates the basic idea of the proposed approach, from which several merits can be identified, as described in the following.

1. *Utilization of the advantage of conventional (k, n)-threshold secret sharing*—It is seen from Steps 1 through 4 of the above algorithm that the secret sharing scheme proposed by Shamir (1979) can be applied properly to image secret sharing, and the advantage of the (k, n)-threshold function of the scheme can be obtained. The previously-mentioned problem of creating an enormous secret data value caused by direct use of the Shamir method is solved.

2. *Providing steganographic effect*—Step 1 takes the top-leftmost pixel value x_i of each camouflage image block as the x value used in Eq. (1) for computing the corresponding value of $F(x_i)$ in Step 4. The secret share $(x_i, F(x_i))$ can then be kept by a participant as in a conventional approach. However, it is desired further in this study that this share be hidden to reduce the possibility of being stolen or tampered with. It is observed from Step 1 that the data value x_i belongs to the top-leftmost pixel that is just part of a camouflage image. The camouflage image contains a visible content that can be selected arbitrarily by a secret sharing participant, and this gives an effect of disguise to reduce possible suspicion coming from illicit invaders. Furthermore, in Step 5, we also propose to hide, using any data hiding technique, the data of the corresponding value of $F(x_i)$ behind the other three pixels of the camouflage image B_i. This as well creates a steganographic effect because what an invader sees is still the camouflage image itself. This is perhaps one of the most desirable merits in security applications. Note that the visual cryptography schemes proposed so far in literature are mostly lossy and with low quality in nature. We will describe the method we employ for data hiding without creating obvious image changes later in this paper.

3. *Offering authentication capability*—In Step 6, we propose to insert a watermark signal (a bit) in the data of one of the three pixels W_i, V_i, and U_i. The details will be described later. This offers a capability of checking the fidelity of each stego-image before it is utilized in a secret recovery process. This is desirable because sometimes a secret sharing participant might incidentally bring an erroneous stego-image to the secret recovery session, or might even intentionally provide a false image to prevent successful secret recovery. In such cases, an additional authentication process for checking the fidelity of all participants' image data before secret recovery starts is very helpful. Otherwise, it will get no way to find out which participant is interfering the secret recovery process if such a case does occur.

4. *Extensibility of the proposed approach to handle color images*—The above basic algorithm is designed for sharing gray-scale images. However, it is not difficult to see that it can be extended to handle color images: simply choose each camouflage image to be a color one, and apply the above algorithm respectively to each of the color channels of the image (e.g., in each of the R, G, and B channels if an RGB image is used), followed by the step of composing the three resulting stego-images, one for each channel, into a color image for a participant to keep. This should be contrasted with certain existing visual secret sharing methods (Verheul and van Tilborg, 1997; Blundo et al., 2000; Lin and Tsai, 2003) which so far can only deal with black-and-white or gray-scale images, or with color images with only a very few number of colors. Our experimental results show that the hidden data are imperceptible.

5. *Providing more security control*—The above algorithm provides at least three levels of security protection, namely, the (k, n)-threshold secret sharing, the steganographic effect, and the authentication capability, which will prevent attacks or illicit access more effectively. The proposed scheme will thus be useful for many high-security applications.

3.3. Elaboration of Algorithm 1 for the proposed approach

Algorithm 1 is just a sketch of a more detailed one that implements the proposed approach. In the following, we describe its details and the ideas behind them in order to reveal the merits of the proposed approach mentioned above. The detailed algorithm will be given in the next section.

1. *Fulfillment of the requirement of the distinction among the values of x_i*—One of the requirements for the applicability of Eqs. (1)–(4) as shown in Shamir (1979) is that all the values of x_i must be distinct from one another. However, the values of x_i of two or more camouflage image blocks might be the same. In such cases, we have to modify the values of x_i to differentiate them. For this, the way we adopt is to compare the values of all x_i one after another, and if any x_i is found to be identical to a former one, just decrement or increment the current value of x_i by one *in an alternative way* (i.e., perform an increment for an identicalness case followed by a decrement for the next case, and then by an increment, and so on). We call this process *an adjustment for differentiating x_i*. This will change the camouflage image appearance to a nearly invisible degree because the change (+1 or −1) is just a very small portion of the full gray scale of 256.

2. *Restriction of the magnitude of $F(x_i)$*—If the magnitude of the value $F(x_i)$ computed in Algorithm 1 is not restricted in a range, the *data hiding capacity* of the three pixels W_i, V_i, and U_i might not be sufficient to embed the data bits of $F(x_i)$. That is, $F(x_i)$ might include too many bits to be embedded in the three pixels. One way out is to perform a "modulo q" operation on the computed value of $F(x_i)$ where q is an integer of a reasonable magnitude. This will cause the result $[F(x_i)]_{\mathrm{mod}\,q}$ to fall in the range of $[0, q-1]$. In our application for image sharing here, $q = 256$ is a proper choice because the secret value y in Eq. (1) comes from the image pixel value and is in the range of [0, 255].

3. *Restriction on the property of q for the modular operation*—It may be shown from Shamir (1979) that when the "modulo q" operation is applied in computing the value of $F(x_i)$, unless the integer q is a prime number, *ambiguity* in the secret recovery result will arise. That is, the secret value y recovered from solving the equations in (2) might not be unique, and this means that unsuccessful secret recovery might occur. To solve this problem, the best choice of q for our case here is $q = 251$ which is the prime number closest to 256.

4. *Truncation of the gray-scale values and restriction on the values of m_i and y*—To meet the modular arithmetic mentioned previously using the prime number $q = 251$, all the values of x_i, that of y, and the corresponding ones of m_i must also be restricted in the range of [0, 250]. Therefore, we must *slightly change* the gray-scale values of the pixels in each camouflage image block B_i which include those values of x_i in Eq. (1), as well as that of the secret image pixel S which is the value of y in Eq. (1). The way we adopt is to truncate all gray-scale values larger than 250 (i.e., those values of 251 through 255) down to the value of 250. This hopefully will not cause too great image quality changes, because our visual inspection of many images reveals that gray-scale values of 250 through 255 are all "bright" enough so that the human vision capability cannot tell differences among them. On the other hand, for specific applications where discrimination among these gray-scale values is necessary, a solution we propose is to map the original 256-level gray scale into a smaller 251-level one for data sharing, and then to transform the latter back to the former for image restoration and inspection. Additionally, the requirement that all the x_i be distinct still need be obeyed after the truncation operations; that is, the previously-mentioned adjustment process for differentiating x_i must be performed if necessary. Based on the above discussions, we now modify Eq. (1) to meet the modular arithmetic as follows for use later in this paper:

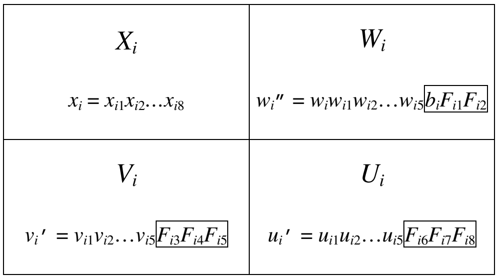
$$F(x) = \left[y + m_1 \times x + m_2 \times x^2 + \cdots + m_{k-1} \times x^{k-1}\right]_{\mathrm{mod}\,q} \quad (1')$$

where $q = 251$, all m_i are selected restrictively to be in the range of [0 250], and y and all x_i are truncated to be 250 if they are larger than 250. In a similar way, we modify Eqs. (2)–(4) by adding the "modulo q" operation to the right-hand side of each of Eqs. (2)–(4). The resulting equations (2′) through (4′) are omitted here.

5. *Hiding the values of $F(x_i)$ by the least significant bit replacement technique*—By the above-mentioned measures, we have restricted the computed values of $F(x_i)$ in the range of [0, 250], and accordingly each $F(x_i)$ can now be represented by a byte, i.e., eight bits. The way we adopt to hide the value of $F(x_i)$ in the camouflage image block B_i is to split the eight bits of $F(x_i)$, denoted by $F_{i1}, F_{i2}, \ldots, F_{i8}$, into three parts consisting of 2, 3, and 3 bits in sequence, and embed them into the values w_i, v_i, and u_i, of the three pixels W_i, V_i, and U_i, respectively, by the least significant bit replacement technique often used in data hiding applications (Lin and Tsai, 2003; Bender et al., 1996; Hsu and Wu, 1999; Wu and Tsai, 1998). More specifically, we hide the first two bits, F_{i1} and F_{i2} of $F(x_i)$ into W_i by replacing the two least significant bits w_{i7} and w_{i8} of W_i with F_{i1} and F_{i2}, respectively, so that the new value of W_i becomes $w_i' = w_{i1}w_{i2}\ldots w_{i6}F_{i1}F_{i2}$. Similarly, the new value of V_i becomes $v_i' = v_{i1}v_{i2}\ldots v_{i5}F_{i3}F_{i4}F_{i5}$ after the three least significant bits of V_i are replaced with the three bits F_{i3}, F_{i4}, and F_{i5} of the second part of $F(x_i)$. Finally, in a similar way the new value of U_i becomes $u_i' = u_{i1}u_{i2}\ldots u_{i5}F_{i6}F_{i7}F_{i8}$.

6. *Embedding the watermark signal for the authentication purpose by the even or odd parity check technique*—To achieve the authentication capability mentioned previously for verifying the fidelity of each stego-image block B_i' before secret recovery, recall that in Step 6 of Algorithm 1 we propose to embed as a watermark signal (a bit) b_i into a pixel of the three pixels W_i, V_i, and U_i of each camouflage image block B_i. Actually, what we do is to take b_i as an *even or odd parity check bit* and embed it into the new data byte w_i' of the second pixel W_i of B_i by replacing the sixth bit w_{i6} of w_i' with b_i so that the resulting data byte of w_i' becomes $w_i'' = w_iw_{i1}w_{i2}\ldots w_{i5}b_iF_{i1}F_{i2}$. Whether b_i is chosen to be 0 or 1 depends on the adopted parity check policy (even or odd) as well as on the number of 1's in w_i'. For example, if $w_i' = 01001100$ and if the *even* parity check policy is adopted, then the check bit b_i, which replaces the bit w_{i6}, is taken to be 0 to make the number of 1's in w_i'' to become even. The parity check policy might even be applied more randomly instead of being fixed, as will be explained later in this paper.

A summary of the above discussions may be illustrated by Fig. 2, which the result of applying Algorithm 1 (including the details described above) to Fig. 1.



X_i	W_i
$x_i = x_{i1}x_{i2}\ldots x_{i8}$	$w_i'' = w_i w_{i1} w_{i2}\ldots w_{i5}\boxed{b_i F_{i1} F_{i2}}$
V_i	U_i
$v_i' = v_{i1}v_{i2}\ldots v_{i5}\boxed{F_{i3}F_{i4}F_{i5}}$	$u_i' = u_{i1}u_{i2}\ldots u_{i5}\boxed{F_{i6}F_{i7}F_{i8}}$

Fig. 2. The result of applying Algorithm 1 to Fig. 1, where $(x_i, F(x_i))$ is the secret share with $F(x_i) = F_{i1}F_{i2}\ldots F_{i8}$ and b_i is the watermark signal bit. The bits framed with rectangles are those changed after applying the algorithm.

4. Proposed process of secret image sharing

Based on Algorithm 1 and the details described in the previous discussions, we can now describe a complete algorithm to implement the proposed secret image sharing approach. Assume that the given secret image is an $m \times m$ gray-scale image. Also, assume that there is a database of camouflage images, each with the size of $2m \times m$. It is suggested to select these camouflage images to be commonly seen pictures, like famous people photographs, beautiful landscape pictures, etc. Such choices of camouflage images will increase the effect of steganography.

Algorithm 2. Detailed process for secret image sharing.

Input : (1) a secret image S with size $m \times m$ to be shared by n participants; (2) a database of more than n camouflage images all with size $2m \times m$; and (3) a secret key K (an integer) for watermark signal generation.

Output : n stego-images in which the secret image and m^2 parity check bits are distributively hidden for sharing and authentication.

Steps:

Step 1. Select n distinct images $I_1, I_2, \ldots, I_n$ from the camouflage image database, each for a secret sharing participant.

Step 2. Use the secret key K as a seed for a pre-selected binary random number generating function f to generate a sequence of m^2 binary numbers $P_1, P_2, \ldots, P_{m^2}$. Regard each binary number P_ℓ ($\ell = 1, 2, \ldots, m^2$) to represent a parity check policy, with $P_\ell = 0$ for the even policy and $P_\ell = 1$ for the odd one.

Step 3. Divide the secret image S into m^2 individual pixels $S_1, S_2, \ldots, S_{m^2}$, with each as a secret image pixel.

Step 4. Divide each camouflage image $I_j (j = 1, 2, \ldots, n)$ into m^2 blocks $B_{j1}, B_{j2}, \ldots, B_{jm^2}$, each of the size of 2×2, and denote the four pixels in each block B_{ji} as X_{ji}, W_{ji}, V_{ji}, and U_{ji}.

Step 5. For each secret image pixel S_i and the corresponding parity check policy P_i, generate a parity check bit b_{ji} (0 or 1) for each $j = 1, 2, \ldots, n$ by taking into consideration of the content of the data byte of W_{ji} in B_{ji} as well as the parity check policy P_i. This results in a sequence of parity bits $b_{1i}, b_{2i}, \ldots, b_{ni}$, with each b_{ji} for use in the ith block B_{ji} in camouflage image I_j.

Step 6. For each secret image pixel S_i, take as input to Algorithm 1 the following data: (1) S_i; (2) $B_{1i}, B_{2i}, \ldots, B_{ni}$; and (3) $b_{1i}, b_{2i}, \ldots, b_{ni}$. The output of the algorithm is a sequence of n 2×2 stego-image blocks $B'_{1i}, B'_{2i}, \ldots, B'_{ni}$, each in a camouflage image.

Step 7. Regard all 2×2 stego-image blocks $B'_{j1}, B'_{j2}, \ldots, B'_{jm^2}$ to compose a $2m \times 2m$ stego-image I'_j as output, and deliver it to the jth participant in the secret sharing group.

In short, the above algorithm may be regarded as an application of Algorithm 1 to each pixel in the given secret image S and collect the resulting 2×2 stego-image blocks to form larger $2m \times 2m$ stego-images. And this is one of the essences of our approach to applying the Shamir method to secret image sharing.

5. Proposed process of secret image recovery

In this section, the proposed secret recovery scheme will be described. Recall that after performing the secret sharing process by Algorithm 2 for a group of n participants, each participant obtains a $2m \times 2m$ stego-image I'_j, $j = 1, 2, \ldots, n$. The proposed secret recovery process is summarized as an algorithm in the following.

Algorithm 3. Process for secret image recovery with stego-image authentication.

Input : (1) a set of at least k *stego-images* I'_j, say t ones, with $k \leqslant t \leqslant n$; and (2) the secret key K used in Algorithm 2 for generating the parity check policies.

Output : a report of failure of secret recovery, or the original secret image S if all the stego-images are authenticated to be genuine.

Steps:

Step 1. Use the secret key K and the binary random number generating function f to generate a sequence of m^2 binary numbers $P_1, P_2, \ldots, P_{m^2}$ which was also generated and used in Algorithm 1 to represent the parity check policies.

Step 2. Divide each stego-image I'_j $(j = 1, 2, \ldots, t)$ into m^2 blocks $B'_{j1}, B'_{j2}, \ldots, B'_{jm^2}$, each of the size of 2×2, and denote the four pixels in each block B'_{ji} as X'_{ji}, W'_{ji}, V'_{ji}, and U'_{ji}.

Step 3. For each stego-image I'_j brought by participant j $(j = 1, 2, \ldots, t)$, perform the following steps for stego-image authentication.

3.1 For each $i = 1, 2, \ldots, m^2$, check all the data bits in the value w'_{ji} of the pixel W'_{ji} of the stego-image block B'_{ji} to see if the number of 1's in them is even or odd. Let P'_{ji} denote the resulting check with P'_{ji} set to be 0 for the even case and to be 1 for the odd case.

3.2 If for $i = 1, 2, \ldots, m^2$, the m^2 values of P'_{ji} are all identical to P_i, then regard the stego-image to pass the authentication and continue; otherwise, decide that the stego-image I'_j has been tampered or is false, stop the algorithm, and report failure of secret recovery.

Step 4. If there are more than k stego-images that have passed the authentication in the last step, then continue; otherwise, stop the algorithm and report failure of secret recovery.

Step 5. For each $i = 1, 2, \ldots, m^2$, perform the following steps to recover the secret image pixel S_i.

5.1 For each $j = 1, 2, \ldots, k$, take the value x_{ji} of the top-leftmost pixel X_{ji} of the 2×2 stego-image block B'_{ji} as a value of x_j appearing in Eq. (4′); extract the data bits of $F(x_{ji})$ from those of the three pixels W_{ji}, U_{ji}, and V_{ji} of B'_{ji}; and take $F(x_{ji})$ as a value of $F(x_j)$ appearing in Eq. (4′).

5.2 Compute, by the use of Eq. (4′), the corresponding value of y as the value s_i for the secret image pixel S_i in terms of the values of all x_j and $F(x_j)$.

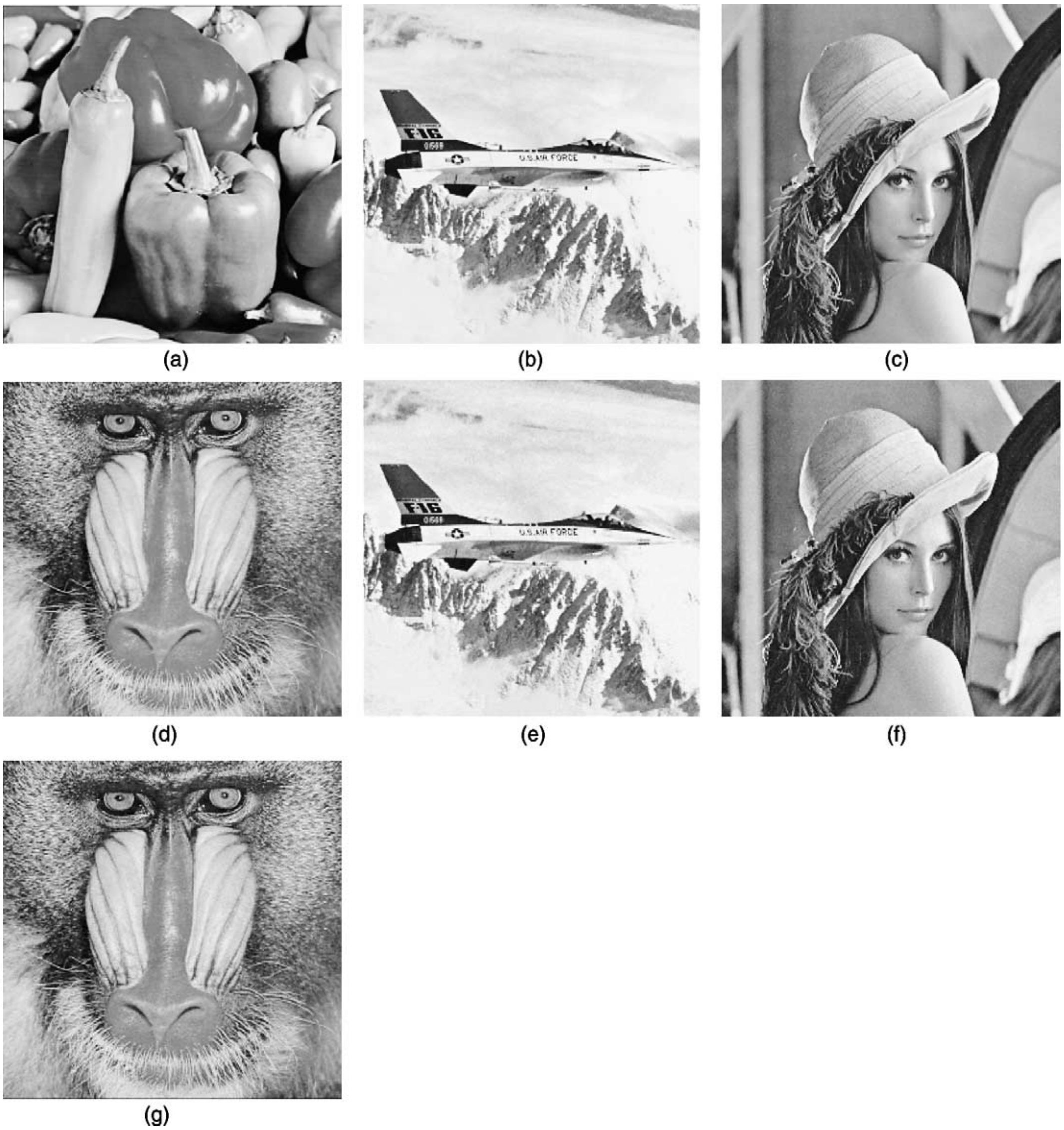


Fig. 3. (a) The secret image. (b) through (d) The camouflage images for participants 1 through 3, respectively (the size of each is four times that of (a)). (e) through (g) The resulting stego-images for participants 1 through 3.

Step 6. Compose all the secret image pixels S_1 through S_{m^2} to form the desired $m \times m$ secret image S as output and stop the algorithm.

6. Experimental results

In this section, some experimental results are shown to prove the feasibility of the proposed scheme. For ease of demonstration, we first use gray-level images to evaluate our scheme. We will show as an example the effect of our scheme for the (2, 3)-threshold case here. At the end of this section, we will give an example of applying our scheme to full color images.

Following the secret sharing process described by Algorithm 2, we first take an image as shown in Fig. 3(a) as the secret image. We then choose three camouflage images arbitrarily and they are shown in Fig. 3(b) through (d). The image size of Fig. 3(a) is 1/4 of those of Fig. 3(b) through (d). After applying Algorithm 2 to the images of Fig. 3(a) through (d), the resulting three stego-images corresponding to Fig. 3(b) through (d) are shown in Fig. 3(e) through (g), respectively. The PSNR values of Fig. 3(e) through (g) are 39.21, 39.16, and 39.16, respectively. The results are satisfactory from the viewpoint of secret hiding effectiveness and stego-image quality.

In the phase of secret recovery, we performed Algorithm 3 to extract the shares first from the stego-images, and then recovered the secret data by using two of the three shares. After getting the secret data, we reconstructed the original secret image pixel by pixel. All the operations were conducted successfully and the original secret image was recovered to be the one shown in Fig. 3(a), as expected.

We have also evaluated the effect of the authentication capability of the proposed scheme. A stego-image in which fragile watermarks were embedded by the scheme is shown in Fig. 4(a). Then we added some modifications to it to simulate image tampering, resulting an image shown in Fig. 4(b). The corresponding result of authentication with the detected tampered image blocks marked in black is shown in Fig. 4(c). It can be observed that all modified regions have been detected correctly.

Finally, we show our experiment results using full color camouflage images. An example of the results is shown in Fig. 5(a) through (g), each of which corresponds to an image in Fig. 3(a) through (g), respectively. And the PSNR values of Fig. 5(e) through 5(g) are 39.07, 39.08, and 39.08, respectively. The stego-images in Fig. 5 (e) through (g) can be inspected to be acceptable visually, as indicated by the reasonable PSNR values.

7. Conclusions

A new scheme for secret image sharing based on the Shamir method (1979) with the additional capabilities of steganography and authentication has been proposed. The proposed scheme has three levels of security protection. First, the (k, n)-threshold function is adopted for a group of n participants to share the secret. Only k or more out of the n shares are collected can the original image data be recovered. Then, the concept of data hiding is employed to embed the shares into camouflage images before delivering the shares to the participants. Finally, the proposed scheme is equipped with the capability of authentication, which can detect false participants' shares before the recovery process is executed. Furthermore, the proposed scheme can also handle full color images, and the quality of the recovery result is nearly lossless. This system is thus suitable for the applications where high security and efficiency is required.

The expansion of the camouflage image size to four times that of the secret image is a weakness of the method. But this is not unique to our method; instead, it is a general problem of most steganographic methods used for hiding data in camouflage images (Bender et al., 1996; Hsu and Wu, 1999; Wu and Tsai, 1998, 1999; Kundur and Hatzinakos, 1999; Adelson, 1990).

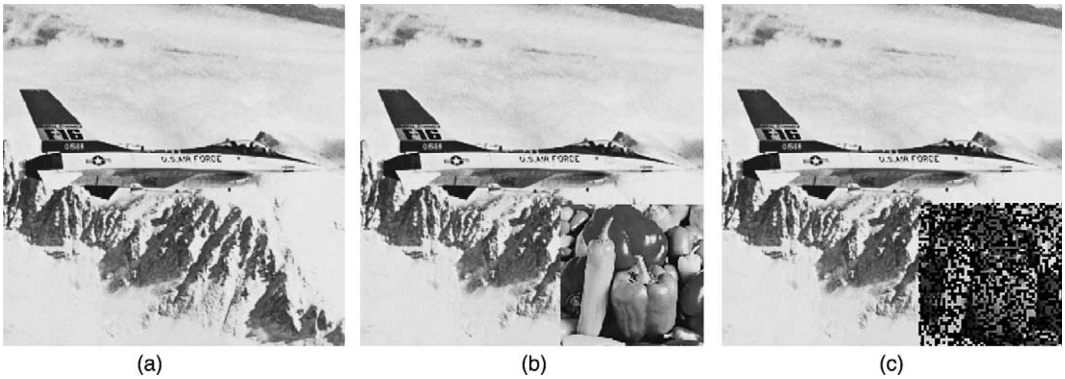


Fig. 4. (a) The stego-image in which fragile watermark signals are embedded. (b) The image with modifications added to (a). (c) The result of authentication.

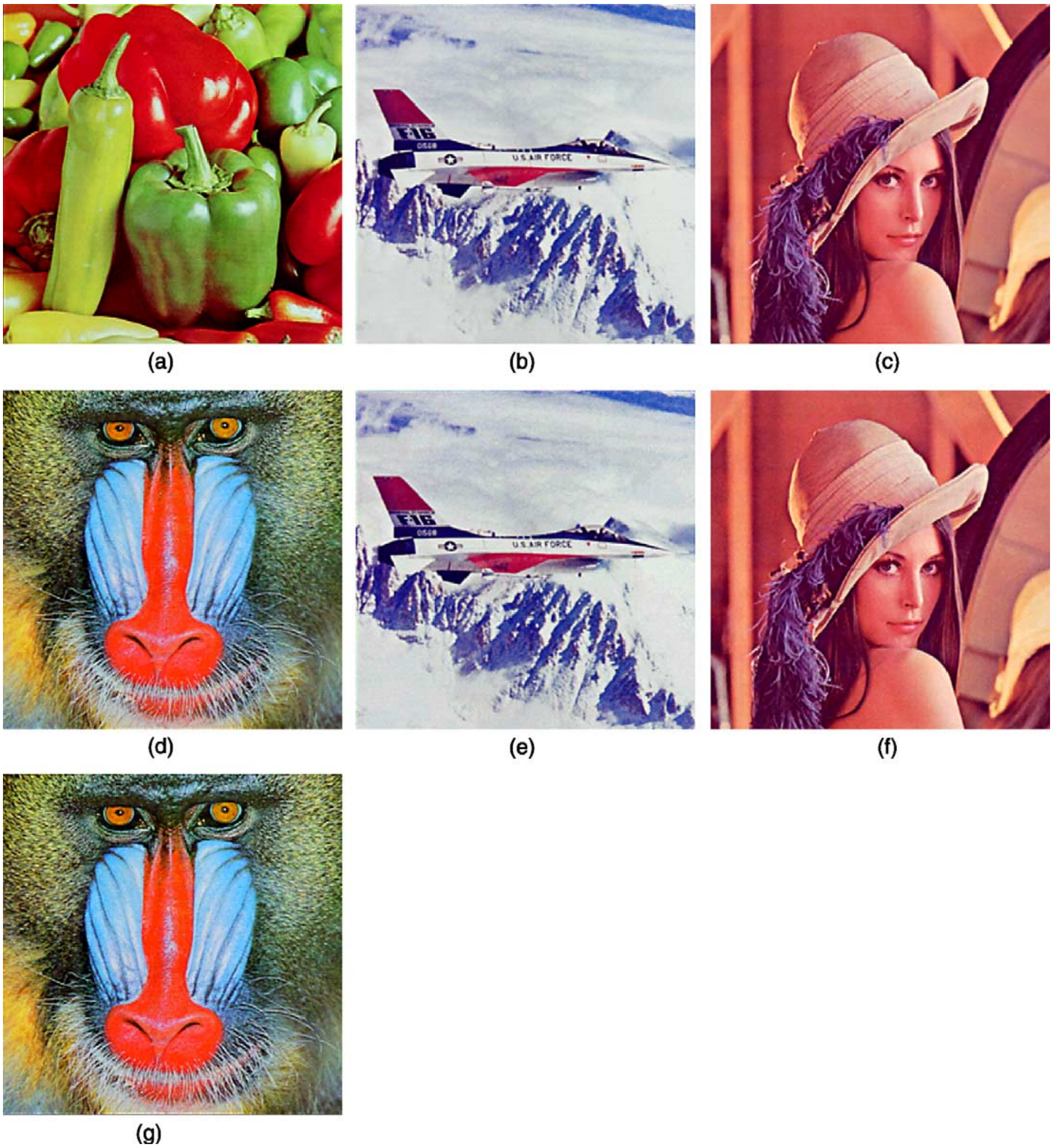


Fig. 5. (a) The secret image. (b) through (d) The camouflage images for participants 1 through 3 (the size of each is four times the size of image of (a)). (e) through (g) The resulting stego-images for participants 1 through 3.

Furthermore, even the conventional visual cryptography methods (Naor and Shamir, 1995; Verheul and van Tilborg, 1997; Blundo et al., 2000; Lin and Tsai, 2003) for secret sharing also have similar share data expansion problems. Nevertheless, the proposed method is still applicable to many situations where the original image is small with the resulting stego-image size still endurable, or where keeping or transmission of the expanded image is not a practical problem.

Acknowledgement

This work was supported partially by the MOE Program for Promoting Academic Excellency of Universities under the grant number 89-1-FA04-1-4.

References

Adelson, E., 1990. Digital signal encoding and decoding apparatus, US Patent no. 4,939,515, 1990.

Bender, W., Gruhl, D., Morimoto, N., Lu, A., 1996. Techniques for data hiding. IBM Systems Journal 35 (3 & 4), 313–336.

Blundo, C., De Santis, A., Naor, M., 2000. Visual cryptography for gray level images. Information Processing Letters 75, 255–259.

Chang, C.C., Lee, H.C., 1993. A new generalized group-oriented cryptoscheme without trusted centers. IEEE Journal on Selected Areas in Communications 11 (5), 725–729.

Hsu, C.T., Wu, J.L., 1999. Hidden digital watermarks in images. IEEE Transactions of Image Processing 8, 58–68.

Kundur, D., Hatzinakos, D., 1999. Digital watermarking for telltale tamper proofing and authentication. Proceedings of the IEEE 87, 1167–1180.

Lin, E.T., Delp, E.J., 1999. A review of fragile image watermarks. Multimedia and Security Workshop in ACM Multimedia '99, Orlando, FL, USA, 1999.

Lin, C.C., Tsai, W.H., 2003. Visual cryptography for gray-level images by dithering techniques. Pattern Recognition Letters 24 (1–3), pp. 349–358.

Naor, M., Shamir, A., 1995. Visual cryptography. In: Advances in Cryptology—EUROCRYPT'94, vol. 950 of Lecture Notes in Computer Science, pp. 1–12.

Shamir, A., 1979. How to share a secret. Communications of the Association for Computing Machinery 22 (11), 612–613.

Sun, H.M., Shieh, S.P., 1994. Construction of dynamic threshold schemes. Electronics Letters 30 (24), 2023–2024.

Verheul, E.R., van Tilborg, H.C.A., 1997. Construction and properties of k out of n visual secret sharing schemes. Designs, Codes, and Cryptography 11, 179–196.

Wu, D.C., Tsai, W.H., 1998. Data hiding in images via multiple-based number conversion and lossy compression. IEEE Transactions on Consumer Electronics 44 (4), 1406–1412.

Wu, D.C., Tsai, W.H., 1999. Embedding of any type of data in images based on a human visual model and multiple-based number conversion. Pattern Recognition Letters 20, 1511–1517.

Chang-Chou Lin was born in Taipei, Taiwan, R.O.C., in 1974. He received the B.S. degree in the Department of Computer Science at National Tsing Hua University in 1996. He works in the Computer Vision Laboratory of the Department of Computer and Information Science at National Chiao Tung University as a research assistant from August 1996, and is currently working toward his Ph.D. degree there. His recent research interests include visual secret sharing, pattern recognition, watermarking, and image hiding.

Wen-Hsiang Tsai was born in Tainan, Taiwan, Republic of China (R. O.C.) in May 10, 1951. He received the B.S. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, Republic of China in 1973, the M.S. degree in Electrical Engineering (with major in Computer Science) from Brown University, Providence, Rhode Island, USA in 1977, and the Ph.D. degree in Electrical Engineering (with major in Computer Engineering) from Purdue university, West Lafayette, Indiana, USA in 1979.

Dr. Tsai joined the faculty of National Chiao Tung University, Hsinchu, Taiwan in November 1979, and stays there until now. He is currently a Professor in the Department of Computer and Information Science and the Vice President of the University. Professor Tsai has been an Associate Professor of the Department of Computer Engineering (now called Department of Computer Science and Information Engineering) and the Acting Director of the Institute of Computer Engineering. In 1984, he joined the Department of Computer and Information Science and acted as the Department Head from 1984 through 1988. He has also been the Associate Director of the Microelectronics and Information System Research Center from 1984 through 1987, the Dean of General Affairs from 1995 to 1996, and the Dean of Academic Affairs of the University from 1999 to 2001. He has served as the Chairman of the Chinese Image Processing and Pattern Recognition Society at Taiwan from 1999 to 2000.

Outside the campus, Professor Tsai has served as a Consultant to several major research institutions in Taiwan. He has acted as the Coordinator of Computer Science in National Science Council, and a member of the Counselor Committee of the Institute of Information Science of Academia Sinica in Taipei. He has been the Editor of several academic journals, including *Computer Quarterly* (now *Journal of Computers*), *Proceedings of the National Science Council*, *Journal of the Chinese Engineers*, *International Journal of Pattern Recognition and Artificial Intelligence*, *Journal of Information Science and Engineering*, and *Pattern Recognition*. He was the Editor-in-Chief of *Journal of Information Science and Engineering* from 1998 through 2000.

Professor Tsai's major research interests include image processing, pattern recognition, computer vision, virtual reality, and information copyright and security protection. So far he has published 257 academic papers, including 107 journal papers and 150 conference papers. He is also granted 6 R.O.C. or USA patents. Dr. Tsai has supervised the thesis studies of 26 Ph.D. students and 101 master students.

Professor Tsai has received many awards, including one Distinguished Research Award, four Outstanding Research Awards, and two Special Research Project Awards, all of the National Science Council in 1987 through 2001. He was the recipient of the 13th Annual Best Paper Award of the Pattern Recognition Society of the USA. He was elected as an Outstanding Talent of Information Science and Technology of the R.O.C. in 1986, received the Best Teacher Award of the Ministry of Education in 1989, and was the recipient of the Distinguished Official Award of the Ministry of Education in 1994. He was the recipient of many Academic Paper Awards made by several academic societies, including two by the Computer Society of the Republic of China in 1989, and thirteen by the Chinese Image Processing and Pattern Recognition Society. He has also received in the past 20 years ten Ph.D. and Master's Thesis Supervision Awards from the Acer Long-Term Foundation, the Xerox Taiwan Company, the Federation of Image Product Companies, the Electrical Engineers Society at Taiwan, and the Information Science Society at Taiwan.

Dr. Tsai is a senior member of the IEEE of the USA, and a member of the Chinese Image Processing and Pattern Recognition Society, the Medical Engineering Society of the Republic of China, and the International Chinese Computer Society.