Fast-weighted secret image sharing

Sian-Jheng Lin
Lee Shu-Teng Chen
Ja-Chen Lin
National Chiao Tung University
Department of Computer Science and Information Engineering
1001 Ta Hsueh Road
Hsinchu, 300
Taiwan
E-mail: stlee@cs.nctu.edu.tw

Abstract. Thien and Lin [*Comput. and Graphics* **26**(5), 765–770 (2002)] proposed a threshold scheme to share a secret image among *n* shadows: any *t* of the *n* shadows can recover the secret, whereas *t*−1 or fewer shadows cannot. However, in real life, certain managers probably play key roles to run a company and thus need special authority to recover the secret in managers' meeting. (Each manager's shadow should be more powerful than an ordinary employee's shadow.) In Thien and Lin's scheme, if a company has less than *t* managers, then manager's meeting cannot recover the secret, unless some managers were given multiple shadows in advance. But this compromise causes managers inconvenience because too many shadows were to be kept daily and carried to the meeting. To solve this dilemma, a weighted sharing method is proposed: each of the shadows has a weight. The secret is recovered if and only if the total weights (rather than the number) of received shadows is at least *t*. The properties of GF(2^r) are utilized to accelerate sharing speed. Besides, the method is also a more general approach to polynomial-based sharing. Moreover, for convenience, each person keeps only one shadow and only one shadow index. *© 2009 Society of Photo-Optical Instrumentation Engineers.* [DOI: 10.1117/1.3168644]

Subject terms: secret image sharing; Galois field; Lagrange polynomial; Chinese remainder theorem.

Paper 080900R received Nov. 17, 2008; revised manuscript received Apr. 30, 2009; accepted for publication May 26, 2009; published online Jul. 15, 2009.

1 Introduction

Blakley[1] and Shamir[2] first proposed the secret sharing scheme in 1979, independently. In their (t,n) threshold scheme, a dealer distributed a secret number into n shadows and each of n participants held one shadow. The secret number could be reconstructed if at least t of the n shadows were received. On the other hand, the secret number could not be revealed if any of $t-1$ or less of the n shadows were received. Later, Shamir[2] introduced the concept of weighted secret sharing in his seminal work. In Shamir's weighted secret sharing with the (t,n) threshold scheme, each of the n participants is assigned with a positive integer weight w_i, where $i=1,2,\ldots,n$ and $1\leqslant w_i\leqslant t-1$. Then, the dealer distributed a secret number into $\Sigma_{i=1}^{n} w_i$ shadows, and the number of shadows that each participant held was equal to their corresponding weight value. The secret could be reconstructed if the sum of the weights of the received participants is no less than the threshold t.

When the secret data are a secret image rather than a secret number, using Blakley's or Shamir's (t,n) threshold scheme to share the secret image will waste much memory space because the size of the secret image is usually very large. To reduce the memory space, Thien and Lin[3] proposed the secret image–sharing method derived from Shamir's scheme;[2] Tso[4] proposed the secret image–sharing method based on Blakley's scheme.[1] In Ref. 3 and 4, the size of each shadow is smaller than that of the secret image. In addition, based on Thien and Lin's secret image–sharing method, the progressive secret image-sharing schemes[5–7] were proposed in succession.

When it comes to the issue of secret image sharing among the weighted participants, based on the concept of Shamir's seminal work,[2] Thien and Lin's method[3] can be simply applied to solve this problem. However, to further improve the execution time in the weighted secret image–sharing phase, a fast–weighted secret image–sharing method is proposed in this paper.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 describes the details of the proposed fast-weighted secret image–sharing scheme. Section 4 shows the experimental results, comparisons, and security analysis. Finally, Sec. 5 draws the conclusions.

2 Related Works

Section 2.1 introduces Thien and Lin's sharing method,[3] and Sec. 2.2 introduces Galois field, which will be utilized in this paper.

2.1 *Thien and Lin's Secret Image–Sharing Method*[3]

In the sharing phase of Thien and Lin's (t,n) threshold method, for each nonoverlapping t pixels of the secret image, the corresponding polynomial is defined as

$$f(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1}(\mathrm{mod}\ p), \tag{1}$$

where $a_0, a_1,\ldots,a_{t-1}$ are the gray values of each t pixels, and p is a prime number. Then,

0091-3286/2009/$25.00 © 2009 SPIE

$$f(1), f(2), \ldots, f(n) \tag{2}$$

are evaluated and assigned to the n shadows sequentially. After processing all pixels in the secret image, the n shadows are thus generated. Because each t pixels in secret image only contributes one pixel to each generated shadow, the size of which is $1/t$ of the secret image.

As for the revealing phase, when any t of the n shadows are received, the first not-yet-used pixel from each of the t shadows is taken, and these t pixels can be used to solve the coefficients $a_0, a_1, \ldots, a_{t-1}$ in Eq. (1) by using Lagrange's polynomial. After sequentially processing all pixels of the t shadows, the secret image can be obtained.

2.2 *Galois Field*

Galois field is a finite field that contains β^r elements, where β is a prime number, and r is a positive integer. (Thien and Lin[3] used $\beta=251$ and $r=1$, but we use $\beta=2$ and $r=8$ in our method.) A finite field also equips with two operators: addition $(+)$ and multiplication $(\cdot)$. Both operators must satisfy the commutative, associative, and distributive laws. The manipulation of addition and multiplication over GF(2^r) are introduced below. Before doing GF(2^r) arithmetic, an r-degree binary-coefficient polynomial $m(X)$, called primitive polynomial, has to be defined first. Primitive means that $m(X)$ has a root α, and $\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^r-2}\}$ are all elements in GF(2^r). [The polynomial $m(X)$ must satisfy certain requirements specified in Galois field GF(2^r), see Lin and Costello,[8] for details. Here, we will use $r=8$ and $m(X)=1+X^2+X^3+X^4+X^8$ in our experiments.]

Let A and B be any two elements in GF(2^r). Then define the addition operator as

$$A+B=A\oplus B,$$

where $\oplus$ is the exclusive-X-OR (XOR) operator. The multiplication operator is somewhat more complicated. Before doing multiplication, convert the two elements A and B to two binary polynomials

$$A=(a_{r-1}\ldots a_2a_1a_0)_2 \rightarrow a_0+a_1X+a_2X^2+\cdots+a_{r-1}X^{r-1}$$

$$B=(b_{r-1}\ldots b_2b_1b_0)_2 \rightarrow b_0+b_1X+b_2X^2+\cdots+b_{r-1}X^{r-1}.$$

Then do the following polynomial multiplication and modulus operations:

$$\begin{aligned}&[(a_0+a_1X+a_2X^2+\cdots+a_{r-1}X^{r-1})(b_0+b_1X+b_2X^2+\cdots\\&\quad+b_{r-1}X^{r-1})]\mathrm{mod}\; m(X)\\&=[(a_0\,\hat{}\,b_0)+(a_0\,\hat{}\,b_1\oplus a_1\,\hat{}\,b_0)X+(a_0\,\hat{}\,b_2\oplus a_1\,\hat{}\,b_1\\&\quad\oplus a_2\,\hat{}\,b_0)X^2+\cdots+(a_{r-1}\,\hat{}\,b_{r-1})X^{2r-2}]\mathrm{mod}\; m(X)\\&=c_0+c_1X+c_2X^2+\cdots+c_{r-1}X^{r-1},\end{aligned}$$

where $\hat{}$ is the AND operator. Finally, the result for $A\cdot B$ can be obtained by

$$A\cdot B=C=(c_{r-1}\ldots c_2c_1c_0)_2.$$

Remark: In general, there exist other definitions for addition and multiplication operators. [The details about GF(2^r) can see be found in Ref. 8.] But we will use the above definition for addition and multiplication throughout the paper.

3 Proposed Fast-Weighted Secret Image–sharing Method

This section has three subsections: Section 3.1 is for weighted secret image sharing; Sec. 3.2 is for weighted secret image revealing; and Sec. 3.3 shows the improved weighted secret image–sharing algorithm based on GF(2^r).

3.1 *Weighted Secret Image–Sharing Phase*

According to the Chinese remainder theorem for polynomials, when we divide

$$h(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1}$$

by a factor $(x-i)$, the remainder is $h(i)$. In symbols,

$$h(i)=h(x)\mathrm{mod}(x-i).$$

Now, when we apply mod p on both sides, we have

$$f(i)=h(i)\mathrm{mod}\;p=[h(x)\mathrm{mod}(x-i)]\mathrm{mod}\;p,$$

where $f(i)=h(i)\mathrm{mod}\;p$ is due to the equation $f(x)=[a_0+a_1x+\cdots+a_{t-1}x^{t-1}]$ mod $p=h(x)$ mod p defined in Eq. (2). Therefore, in Galois field GF(p), i.e., in the field of mod p, we may say that $f(i)$ and $[h(x)\mathrm{mod}(x-i)]$ are equal. In symbols,

$$\begin{aligned}f(i)&=h(x)\mathrm{mod}(x-i)\\&=[a_0+a_1x+\cdots+a_{t-1}x^{t-1}]\mathrm{mod}(x-i)\end{aligned} \tag{3}$$

in Galois field GF(p). That is to say, if we divide that polynomial $a_0+a_1x+\cdots+a_{t-1}x^{t-1}$ by $(x-i)$, then the remainder is a number. If we divide this number by p further, then we obtain $f(i)$. In this paper, to define our own formula of the weighted secret image sharing with the (t,n) threshold scheme, we extend Eq. (2) as

$$g_i^{w_i}(x)=[a_0+a_1x+\cdots+a_{t-1}x^{t-1}]\mathrm{mod}(x-i)^{w_i}, \tag{4}$$

where w_i is the shadow weight and $i=1,2,\ldots,n$. Also, rather than explaining Eq. (4) over GF(251) that Thien and Lin[3] used, we explain Eq. (4) over Galois field GF(2^r). As stated in Sec. 2.2, r is a positive integer and we will use GF(2^8) in our experiments.

Before sharing each nonoverlapping t pixels of the secret image using weighted secret image sharing with (t,n) threshold scheme, the secret image is encrypted first. Next,

$$g_1^{w_1}(x), g_2^{w_2}(x), \ldots, g_n^{w_n}(x) \tag{5}$$

are computed using Eq. (4). Then, the w_i coefficients of the polynomial $g_i^{w_i}(x)$ in order of decreasing power of x are sequentially assigned to the corresponding shadow h_i. After processing all pixels in the secret image, the n shadows $\{(h_1,w_1),(h_2,w_2),\ldots,(h_n,w_n)\}$ are generated. Because t pixels in secret image contribute w_i pixels to the generated shadow h_i, the size of which is w_i/t of the secret image.

In the proposed (t,n) threshold weighted secret image–sharing scheme, the two values index i and weight w_i of the generated shadow h_i are needed for revealing the secret image where $1 \leqslant i \leqslant n$. Like Thien and Lin's method,[3] the value i can be attached to the head of the shadow h_i. As for the value w_i, it can be either simply attached to the head of the shadow h_i or calculated by the size of the shadow. Let the size of the secret image be $|S|$ and the size of shadow be $|h_i|$. Then, the weight w_i can be calculated by the formula

$$w_i = \frac{|h_i|}{\lceil |S|/t \rceil}. \tag{6}$$

3.2 *Weighted Secret Image-Revealing Phase*

If someone gets any m of the n shadows and the sum of the weights of the m shadows is greater than or equal to the threshold t, then the secret image can be recovered. Without loss of generality, let these m shadows be $\{(\tilde{h}_{k_1}, w_{k_1}), (\tilde{h}_{k_2}, w_{k_2'}), \cdots, (\tilde{h}_{k_m}, w_{k_m})\}$, and $\Sigma_{j=1}^{m} w_{k_j} \geqslant t$. Then for each shadow $\tilde{h}_{k_j}$, the first w_{k_j} not-yet-used pixels are sequentially taken and then assigned to the coefficients of polynomial $\tilde{g}_{k_j}^{w_{k_j}}(x)$ in order of decreasing power of x. After obtaining $\tilde{g}_{k_j}^{w_{k_j}}(x)$, we have

$$\tilde{g}_{k_j}^{w_{k_j}}(x) = \tilde{f}(x) \bmod (x-k_j)^{w_{k_j}}, \tag{7}$$

where $j=1,2,\ldots,m$. Because the m divisors $(x-k_1)^{w_{k_1}}$, $(x-k_2)^{w_{k_2}}, \ldots, (x-k_m)^{w_{k_m}}$ in Eq. (7) are pairwise relatively prime, as stated in Ref. 9, $\tilde{f}(x)$ can be solved using extended Lagrange interpolation as

$$\tilde{f}(x) = \sum_{j=1}^{m} \left(\tilde{g}_{k_j}^{w_{k_j}}(x) u_j(x) \bmod (x-k_j)^{w_{k_j}} \prod_{\substack{l=1 \\ l \neq j}}^{m} (x-k_l)^{w_{k_l}} \right), \tag{8}$$

where

$$u_j(x) = \left(\prod_{\substack{l=1 \\ l \neq j}}^{m} (x-k_l)^{w_{k_l}} \right)^{-1} \bmod (x-k_j)^{w_{k_j}}.$$

In addition, according to the Chinese remainder theorem for polynomials, $\tilde{f}(x)$ is a unique polynomial with degree less than $\Sigma_{j=1}^{m} w_{k_j}$. Because $\Sigma_{j=1}^{m} w_{k_j} \geqslant t$, the polynomial $\tilde{f}(x)$ is identical to the original polynomial $f(x)$, where the degree of $f(x)$ is less than t. In other words, the t coefficients $a_0, a_1, \ldots, a_{t-1}$ in Eq. (4) can be obtained.

After sequentially processing all pixels of the m shadows, the encrypted secret image can be reconstructed. The encrypted secret image is then decrypted to obtain the secret image.

3.3 *Fast-Weighted Secret Image–Sharing Algorithm*

The computing time of Eq. (5) is improved by using the properties of GF(2^r). The utilized property is that the additive inverse of an element over GF(2^r) is the element itself. In other words,

$$x = -x. \tag{9}$$

By Eq. (9), the following equation is derived:

$$\begin{aligned} (x-u)^2 = (x+u)^2 &= x^2 + xu + xu + u^2 \\ &= x^2 + xu - xu + u^2 = x^2 + u^2. \end{aligned} \tag{10}$$

Then, Eq. (10) can be extended as

$$\begin{aligned} (x-u)^{2^q} = (x+u)^{2^q} &= (x^2+u^2)^{2^{q-1}} = (x^4+u^4)^{2^{q-2}} \\ &= \cdots = x^{2^q} + u^{2^q}, \end{aligned} \tag{11}$$

where q is a positive integer and u is an element in GF(2^r). Let $\Sigma_{j=0}^{2^q-1} a_j x^j$ and $(x^{2^{q-1}} + u^{2^{q-1}})$ be two polynomials. Then $\Sigma_{j=0}^{2^q-1} a_j x^j$ is divided by $(x^{2^{q-1}} + u^{2^{q-1}})$ over GF(2^r) to get the quotient $a_{2^q-1} x^{2^{q-1}-1} + a_{2^q-2} x^{2^{q-1}-2} + \cdots + a_{2^{q-1}}$ and the remainder $(a_{2^{q-1}_{-1}} + a_{2^q-1} u^{2^{q-1}}) x^{2^{q-1}-1} + (a_{2^{q-1}_{-2}} + a_{2^q-2} u^{2^{q-1}}) x^{2^{q-1}-2} + \cdots + (a_0 + a_{2^{q-1}} u^{2^{q-1}})$. By Eq. (11), we have $x^{2^{q-1}} + u^{2^{q-1}} = (x+u)^{2^{q-1}}$. Therefore, $\Sigma_{j=0}^{2^q-1} a_j x^j$ can be expressed as

$$\begin{aligned} \sum_{j=0}^{2^q-1} a_j x^j = {} & (a_{2^q-1} x^{2^{q-1}-1} + a_{2^q-2} x^{2^{q-1}-2} + \cdots + a_{2^{q-1}})(x \\ & + u)^{2^{q-1}} + (a_{2^{q-1}_{-1}} + a_{2^q-1} u^{2^{q-1}}) x^{2^{q-1}-1} + (a_{2^{q-1}_{-2}} \\ & + a_{2^q-2} u^{2^{q-1}}) x^{2^{q-1}-2} + \cdots + (a_0 + a_{2^{q-1}} u^{2^{q-1}}). \end{aligned} \tag{12}$$

However, if one uses Eq. (12) for sharing directly, the weight w_i is restricted as a power of two (2^{q-1}). In order to achieve a generalized version, a recursive algorithm is proposed below. Let $\hat{i} \leftarrow i$, $\hat{w}_i \leftarrow w_i$, $\hat{t} \leftarrow 2^{\lceil \log_2 t \rceil}$, and $\hat{f}(x) \leftarrow f(x)$. Now, $\hat{f}(x) \bmod (x+\hat{i})^{\hat{w}_i}$ is solved using the following recursive algorithm $A[\hat{f}(x), \hat{i}, \hat{w}_i, \hat{t}]$.

Algorithm 1 (Fast–weighted secret image–sharing algorithm $A[\hat{f}(x), \hat{i}, \hat{w}_i, \hat{t}]$):

Input: A polynomial $\hat{f}(x)$, three positive integers $\hat{i}$ (index), $\hat{w}_i$ (weight), and $\hat{t}$ [a value in $\{1, 2, 2^2, 2^3, \ldots\}$, and $\hat{t}$ is the number of polynomial coefficients for $\hat{f}(x)$].

Output: The shadow values with index $\hat{i}$ and weight $\hat{w}_i$ [the coefficients of $\hat{f}(x) \bmod (x+\hat{i})^{\hat{w}_i}$].

Step 1. According to Eq. (12), rewrite $\hat{f}(x)$ as $\hat{f}(x) = \hat{Q}(x)(x+\hat{i})^{\hat{t}/2} + \hat{R}(x)$, where $\hat{Q}(x)$ and $\hat{R}(x)$ are, respectively, the quotient and the remainder on dividing $\hat{f}(x)$ by $(x+\hat{i})^{\hat{t}/2} = x^{\hat{t}/2} + \hat{i}^{\hat{t}/2}$ over GF(2^r).

Step 2. Compare $\hat{w}_i$ with $\hat{t}/2$.

Case 1. If $\hat{w}_i=\hat{t}/2$, then $\hat{f}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}=[\hat{Q}(x)(x+\hat{i})^{\hat{t}/2}+\hat{R}(x)]\text{mod}(x+\hat{i})^{\hat{w}_i}=[\hat{Q}(x)(x+\hat{i})^{\hat{t}/2}+\hat{R}(x)]\text{mod}(x+\hat{i})^{\hat{t}/2}=\hat{R}(x)$. Therefore, return $\hat{R}(x)$.

Case 2. If $\hat{w}_i<\hat{t}/2$, then $\hat{f}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}=[\hat{Q}(x)(x+\hat{i})^{\hat{t}/2}+\hat{R}(x)]\text{mod}(x+\hat{i})^{\hat{w}_i}=\hat{R}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}$. Then, $\hat{R}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}$ is recursively computed by $A[\hat{R}(x),\hat{i},\hat{w}_i,\hat{t}/2]$. Finally, return $\hat{R}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}$.

Case 3. If $\hat{w}_i>\hat{t}/2$, then $\hat{f}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}=[\hat{Q}(x)(x+\hat{i})^{\hat{t}/2}+\hat{R}(x)]\text{mod}(x+\hat{i})^{\hat{w}_i}=[\hat{Q}(x)\text{mod}(x+\hat{i})^{\hat{w}_i-\hat{t}/2}](x+\hat{i})^{\hat{t}/2}+\hat{R}(x)$.

From Eq. (11), because $(x+\hat{i})^{\hat{t}/2}=x^{\hat{t}/2}+i^{\hat{t}/2}$, $\hat{f}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}=[\hat{Q}(x)\text{mod}(x+\hat{i})^{\hat{w}_i-\hat{t}/2}](x+\hat{i})^{\hat{t}/2}+\hat{R}(x)=[\hat{Q}(x)\text{mod}(x+\hat{i})^{\hat{w}_i-\hat{t}/2}]\times(x^{\hat{t}/2}+\hat{i}^{\hat{t}/2})+\hat{R}(x)$. Then, $\hat{Q}(x)\text{mod}(x+\hat{i})^{\hat{w}_i-\hat{t}/2}$ is recursively computed by $A[\hat{Q}(x),\hat{i},\hat{w}_i-\hat{t}/2,\hat{t}/2]$. Finally, return $[\hat{Q}(x)\text{mod}(x+\hat{i})^{\hat{w}_i-\hat{t}/2}](x^{\hat{t}/2}+\hat{i}^{\hat{t}/2})+\hat{R}(x)$.

Notably, the above algorithm can be abbreviated as a recursive function. Let $\hat{i}=i$, $\hat{w}_i=w_i$, $\hat{t}=2^{\lceil\log t\rceil}$ and $\hat{f}(x)=\Sigma_{i=t}^{\hat{t}}0x^i+f(x)$. Then,

$$A[\hat{f}(x),\hat{i},\hat{w}_i,\hat{t}]=A[\hat{Q}(x)(x+\hat{i})^{\hat{t}/2}+\hat{R}(x),\hat{i},\hat{w}_i,\hat{t}]=\begin{cases}\text{Case 1: } \hat{R}(x), & \text{if } \hat{w}_i=\hat{t}/2,\\ \text{Case 2: } A(\hat{R}(x),\hat{i},\hat{w}_i,\hat{t}/2), & \text{if } \hat{w}_i<\hat{t}/2,\\ \text{Case 3: } A(\hat{Q}(x),\hat{i},\hat{w}_i-\hat{t}/2,\hat{t}/2)(x^{\hat{t}/2}+\hat{i}^{\hat{t}/2})+\hat{R}(x), & \text{if } \hat{w}_i>\hat{t}/2.\end{cases}$$

Now, for the recursive function above, an example is given below.

3.4 Demonstration Example of Fast-Weighted Secret Image Sharing

3.4.1 Input of the demonstration

1. A polynomial $f(x)=2x^5+5x^4+2x^3+6x^2+3x+1$ whose coefficients are all in $\text{GF}(2^3=8)$ (i.e., all in the range {0, 1, 2, 3, 4, 5, 6, 7})
2. A shadow index $i=1$, a shadow weight $w_i=5$, and a threshold $t=6$

3.4.2 Demonstration purpose

Show how to compute the corresponding shadow value $g_1^5(x)=\hat{f}(x)\text{mod}(x+\hat{i})^{\hat{w}_i}=(0x^7+0x^6+2x^5+5x^4+2x^3+6x^2+3x+1)\text{mod}(x+1)^5$ where $\hat{i}=i=1$, $\hat{w}_i=w_i=5$, and $\hat{f}(x)=\Sigma_{i=t}^{\hat{t}}0x^i+f(x)=0x^7+0x^6+2x^5+5x^4+2x^3+6x^2+3x+1$ is the whole-power-of-two version of f (by adding the missing zero coefficients to f so that all $\hat{t}=2^{\lceil\log_2 t\rceil}=2^{\lceil\log_2 6\rceil}=8$ coefficients appear).

Fig. 1 The 512×512 secret image Lena.

3.4.3 Demonstration detail

According to the recursive function of our sharing algorithm, we have

$$\begin{aligned}A(\hat{f}(x),\hat{i},\hat{w}_i,\hat{t}) &= A(0x^7+0x^6+2x^5+5x^4+2x^3+6x^2+3x\\ &\quad +1,1,5,8)\\ &= A[(0x^3+0x^2+2x+5)(x+1)^4+(2x^3+6x^2\\ &\quad +1x+4),1,5,8] \quad [\because \text{Eq. (12)}]\\ &= A[0x^3+0x^2+2x+5,1,1,4](x^4+1^4)+(2x^3\\ &\quad +6x^2+1x+4)\\ &\quad (\because \hat{w}_i=5>\hat{t}/2=4, \ \therefore \ \text{case 3})\\ &= A[(0x+0)(x+1)^2+(2x+5),1,1,4](x^4\\ &\quad +1^4)+(2x^3+6x^2+1x+4) \quad [\because \text{Eq. (12)}]\\ &= A(2x+5,1,1,2)(x^4+1^4)+(2x^3+6x^2+1x\\ &\quad +4) \quad (\because \hat{w}_i=1<\hat{t}/2=2, \ \therefore \ \text{case 2})\\ &= A[2(x+1)^1+7,1,1,2](x^4+1^4)+(2x^3+6x^2\\ &\quad +1x+4) \quad [\because \text{Eq. (12)}]\\ &= 7(x^4+1^4)+(2x^3+6x^2+1x+4)\\ &\quad (\because \hat{w}_i=1=\hat{t}/2=1, \ \therefore \ \text{case 1})\\ &= 7x^4+2x^3+6x^2+1x+3.\end{aligned}$$

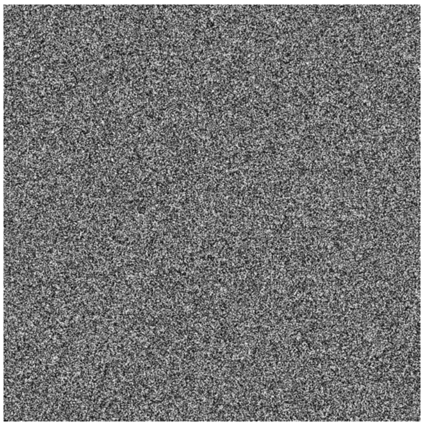

Fig. 2 Encrypted image of Lena.

Therefore, $g_1^5(x)=(2x^5+5x^4+2x^3+6x^2+3x+1)\mathrm{mod}(x+1)^5 =A(0x^7+0x^6+2x^5+5x^4+2x^3+6x^2+3x+1,1,5,8)=7x^4+2x^3+6x^2+1x+3$.

4 Experimental Results, Comparisons, and Security Analysis

Section 4.1 shows the experimental results. Section 4.2 compares our method to Thien and Lin's method.[3] Section 4.3 is a discussion about the security of our method.

4.1 *Experimental Results*

The standard 512 × 512 gray-level image Lena is shown in Fig. 1, which is used as the secret image in the experiments. Figure 2 shows the encrypted image of Fig. 1; the encryption uses exclusive-OR operation between a random sequence and the gray values of the secret image. Then, the proposed fast-weighted secret image sharing with $(t=256, n=7)$ threshold scheme over GF(2^8) is used to share the encrypted secret image (Fig. 2), and $n=7$ shadows are thus generated and shown in Figs. 3(a)–3(g) with the shadow weight 160, 64, 24, 8, 134, 12, and 3, respectively. Figure 4 is the image revealed by Figs. 3(a)–3(d), and the revealed image is identical to Fig. 1.

Figure 5 compares the execution time in the weighted secret image–sharing phase using Thien and Lin's $(t=256, n=w_i)$ threshold scheme[3] and our $(t=256, n=1)$ threshold scheme. The two schemes are both tested on an AMD Athlon 3500+ computer with 3GB of RAM. Notably, the execution time of our sharing algorithm is 7 ± 3 ms for each of these 255 sets of weights, whereas the execution time increases linearly as the weight value increases in Thien and Lin's direct and repeated application (using multiple shadows to simulate weighted feature).

4.2 *Comparisons to Thien and Lin's Scheme*[3]

Some advantages of our method are presented in this section (compared to Thien and Lin's method[3]).

4.2.1 *Time complexity*

The time complexity of the weighted secret image sharing using Thien and Lin's scheme[3] and our scheme is analyzed as follows. For Thien and Lin's (t,n) threshold scheme, when sharing each nonoverlapping t pixels of the secret image among $\Sigma_{i=1}^{n} w_i$ shadows, based on Shamir's[2] seminal work, $f(1), f(2), \ldots, f(\Sigma_{i=1}^{n} w_i)$ are computed by Eq. (1). Because there are t multiplications and $(t-1)$ additions in Eq. (1), the time complexity of sharing secret image with size $|S|$ among $\Sigma_{i=1}^{n} w_i$ shadows by Thien and Lin's scheme is $\theta(\Sigma_{i=1}^{n} w_i) \times \theta(t) \times \theta(|S|/t) = \theta(|S| \times \Sigma_{i=1}^{n} w_i)$. Because $\Sigma_{i=1}^{n} w_i < nt$, we have $\theta(|S|\Sigma_{i=1}^{p} w_i)=O(|S|nt)$.

As for our scheme, when sharing each nonoverlapping t pixels of the secret image among n shadows, $f(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1}$ in Eq. (4) is expanded to $f(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1}+0x^t+\cdots+0x^{2^{\lceil \log_2 t \rceil}}$ if the value of t is not power of 2. Then, $g_1^{w_1}(x), g_2^{w_2}(x), \ldots, g_n^{w_n}(x)$ in Eq. (5) are computed using our fast-weighted secret image-sharing algorithm. Suppose the time complexity of computing each

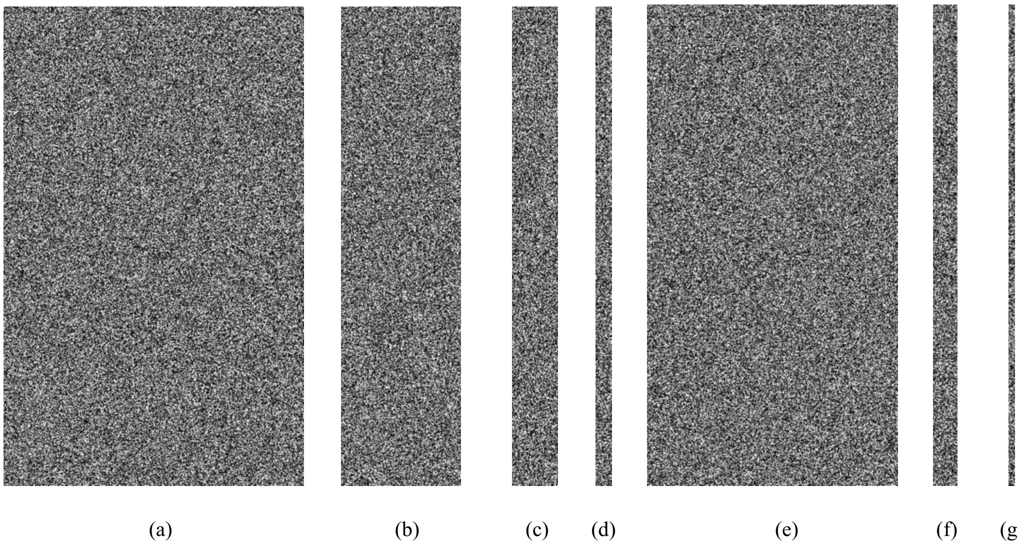

Fig. 3 Seven generated shadows with the shadow weight (a) 160, (b) 64, (c) 24, (d) 8, (e) 134, (f) 12, and (g) 3.

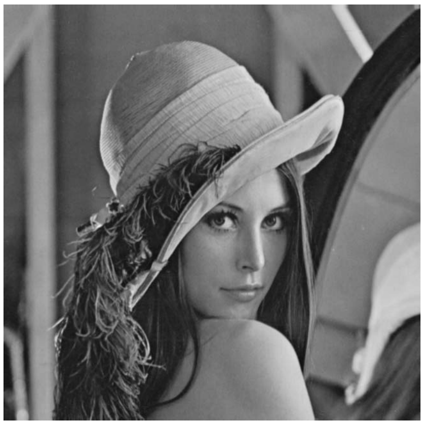

Fig. 4 Image revealed from Figs. 3(a)–3(d).

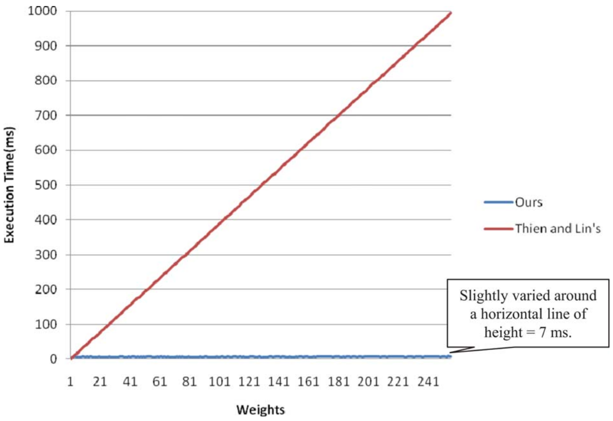

Fig. 5 Execution time in the weighted secret image-sharing using Thien and Lin's (t=256, n=w_i) threshold scheme[3] and our (t=256, n=1) threshold scheme.

$g_i^{w_i}(x)$ in Eq. (5) is $T(\hat{t})$, where $i=1,2,\cdots,n$. Because the concept of the recursive function is applied in our algorithm, and there are $\hat{t}/2$ multiplications and $\hat{t}/2$ additions in the step 1 of Algorithm 1, the recurrence relation $T(\hat{t})=T(\hat{t}/2)+\theta(\hat{t})$ can be derived. The recurrence relation is then solved by the substitution method to obtain $T(\hat{t})=\theta(\hat{t})$. Because $\hat{t}\leftarrow 2^{\lceil \log_2 t \rceil}$, the value of $\hat{t}$ is at most two times of t. Therefore, we have $T(\hat{t})=\theta(\hat{t})=\theta(t)$. Thus, the time complexity of sharing secret image with size $|S|$ among n shadows by our scheme is $\theta(n)\times\theta(t)\times\theta(|S|/t)=\theta(|S|n)$.

4.2.2 *More general scheme for polynomial-based sharing*

In our weighted sharing scheme, according to the Chinese remainder theorem for polynomials, the n polynomials $x-1$, $x-2,\ldots,x-n$ in Eq. (3) can be replaced by n other sharing polynomials, such as x^2+x+1, $x^2+x+2,\ldots,x^2+x+n$, as long as these n polynomials are pairwise relatively prime (i.e., no pair of polynomials has a nontrivial common factor). Notably, Thien and Lin's method[3] is only a special case of this generalized scheme (i.e., the n shadows of Thien and Lin's are evaluated by $f(i)=f(x)\text{mod}(x-i)$, for $i=1,2,\ldots,n$. In other words, only $\{x-1,x-2,\ldots,x-n\}$ were used by Thien and Lin,[3] whereas we can use all sharing polynomials that are pairwise relatively prime).

4.2.3 *Better Performance when pixel values are >250*

The computations in Thien and Lin's sharing process are over GF(251). All the gray values 251–255 of the gray-level secret image have to be truncated to 250. Therefore, the recovered secret image is lossy. To recover the secret image without any loss, Thien and Lin[3] introduce a preprocessing to decompose the gray value of >250; for example, 253 is separated as a pair of pixels {250 and 3}. This preprocessing will waste time and slightly increase the size of their shadows. However, because we use GF(256) in our weighted sharing procedure, the secret image can be lossless reconstructed without additional postprocessing.

4.2.4 *Each participant keeps only one index and only one shadow (hence, more convenient and space savings)*

If a company wants to apply Thien and Lin's (t,n) scheme[3] directly to achieve the goal of weighted participants, then the company can let some participants hold multiple shadows. To be more specific, because each shadow generated by Thien and Lin's scheme[3] has weight 1, the participant $i(1\leqslant i\leqslant n)$ whose weight is w_i should be assigned w_i shadows, and each of which will be attached with an index value for the secret-recovery meeting in the future. The w_i indices for participant i will cause an inconvenience, and the w_i shadows (rather than a single shadow) also waste the storage space of participant i. Moreover, if there are three participants whose weights are, respectively, 128, 122, and 99, then, Thien and Lin's method[3] will be in trouble. This is because the first participant will obtain 128 shadows with the 128 indices values being 1, 2, …, 128; and the second participant will obtain 122 shadows with the indices values being 129, 130, …, 250. As for the third participant, there is "no" shadow left for him because GF(251) restricts the input index value be <251; therefore, the system cannot generate >250 shadows. However, by using our method, the first participant will obtain only a shadow with the index value 1 and the weight value 128; the second participant will obtain a shadow with the index value 2 and the weight value 122; and the third participant will obtain a shadow with the index value 1 and the weight value 99. Hence, besides giving convenience to each participant, the proposed method also keeps the storage space of each participant much more economically.

4.3 *Security Analysis*

The security analysis is divided into two parts: (i) a group of shadows with total weights $t-1$ cannot reveal the secret image, and (ii) shadows of different weights are not equally secure.

First, suppose that the m' obtained shadows are $\{(\widetilde{h}_{k_1},w_{k_1}),(\widetilde{h}_{k_2},w_{k_2}),\ldots,(\widetilde{h}_{k_{m'}},w_{k_{m'}})\}$ and the sum of their weights is $t-1$ (i.e., $\Sigma_{j=1}^{m'}w_{k_j}=t-1$), then we analyze the probability of obtaining the secret image by guessing. According the Chinese remainder theorem for polynomials, we can construct a unique polynomial $\tilde{f}'(x)$ where the degree is less than $\Sigma_{j=1}^{m'} w_{k_j} = t-1$ from these m' shadows. After obtaining $\tilde{f}'(x)$, to reveal the $\tilde{f}(x)$ in Eq. (7) by $\tilde{f}'(x)$, we have

$$\tilde{f}(x) = \alpha \prod_{j=1}^{m'} (x-k_j)^{w_{k_j}} + \tilde{f}'(x),$$

where α is a non-negative integer less than $2^8 = 256$ [because GF(2^8) is used in our experiments]. Because there are $2^8 = 256$ possible values of α, the possibility of guessing the right solution $\tilde{f}(x)$ is 1/256. For a 512 × 512 secret image, because there are $512 \times 512/t$ polynomials, the possibility of obtaining the right secret image is $(1/256)^{(512\times 512)/t}$, which is a form similar to the $(1/251)^{(512\times 512)/t}$ given in Thien and Lin's paper[3].

Second, we analyze the probability of obtaining the secret image by using only one shadow. Given a shadow h_{w_i} of weight w_i, then the polynomial $g_i^{w_i}(x)$ can be obtained using the shadow h_{w_i}. Now, to use $g_i^{w_i}(x)$ to reveal the $\tilde{f}(x)$ in Eq. (7), we have $\tilde{f}(x) = Q'(x)(x-i)^{w_i} + g_i^{w_i}(x)$, where $Q'(x)$ is an unknown polynomial with a degree of less than $t-w_i$. Therefore, there is $1/256^{t-w_i}$ chance to find out the polynomial $Q'(x)$ by guessing. On the other hand, there are $512 \times 512/t$ polynomials for a given 512 × 512 secret image; thus, the possibility of finding out the secret image is $(1/256^{t-w_i})^{(512\times 512)/t} = (1/256)^{512\times 512\times[1-(w_i/t)]}$. This shows that shadows of different weights are not equally secure, because the security of each shadow is weight dependent. To find out the secret image by guessing, the owner of a larger-weight shadow has more of a chance than the owner of a smaller weight has. This agrees with our daily-life experience: a higher-ranking manager (having heavier weight) has more of a chance to uncover the company's secret than a lower-ranking employee has.

5 Conclusions

In this paper, a fast-weighted secret image–sharing with (t,n) threshold method is proposed. The method shares the secret image among the weighted participants, and the secret image can be losslessly recovered if the sum of the weights of the participants is greater than or equal to the threshold t. Besides, the execution time in the weighted secret image–sharing phase is improved by using the properties of GF(2^r). As shown in Fig. 5, our execution time is better than that of Thien and Lin[3] when $w_i > 1$. The executives of a company can use our method to share the secret image.

Acknowledgments

This work is supported by National Science Council of Republic of China, under Grant No NSC97-2221-E-009-120-MY3. The authors also thank the reviewers for valuable suggestions.

References

1. G. R. Blakley, "Safeguarding cryptographic keys," *Proc. of AFIPS Natl. Comput. Conf.* Vol. **48**, pp. 313–317 (1979).
2. A. Shamir, "How to share a secret," *Commun. ACM* **22**(11), 612–613 (1979).
3. C. C. Thien and J. C. Lin, "Secret image sharing," *Comput. Graph.* **26**(5), 765–770 (2002).
4. H. K. Tso, "Sharing secret images using Blakley's concept," *Opt. Eng.* **47**(7), 077001 (2008).
5. S. K. Chen and J. C. Lin, "Fault-tolerant and progressive transmission of images," *Pattern Recogn.* **38**(12), 2466–2471 (2005).
6. R. Z. Wang and S. J. Shyu, "Scalable secret image sharing," *Signal Process. Image Commun.* **22**(4), 363–373 (2007).
7. W. P. Fang, "Friendly progressive visual secret sharing," *Pattern Recogn.* **41**(4), 1410–1414 (2008).
8. S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed., Pearson Education, Upper Saddle River, New Jersey (2004).
9. D. Bini and V. Y. Pan, *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms*, Birkhauser, Boston (1994).

Sian-Jheng Lin received his BS and MS in computer science from National Chiao Tung University (NCTU) in 2004 and 2006, respectively. His is currently a PhD candidate in the Computer Science Department of National Chiao Tung University. His current research interests include image processing and secret sharing.

Lee Shu-Teng Chen received his BS in computer science from NCTU, Taiwan, in 1999, and MS in computer science and information engineering from National Taiwan University, Taiwan, in 2001. He has been in the PhD program since 2004 and currently is a PhD candidate in the Department of Computer Science and Information Engineering at NCTU. His current research interests include data hiding and image sharing.

Ja-Chen Lin received his BS and MS from NCTU, Taiwan. In 1988, he received his PhD in mathematics from Purdue University, West Lafayette, Indiana. He joined the Department of Computer and Information Science at NCTU, in 1988, and became a professor there. His research interests include pattern recognition and image processing. Lin is a member of the Phi-Tau-Phi Scholastic Honor Society.