Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 4802–4817.
DOI: 10.3934/mbe.2019242
Received: 24 January 2019
Accepted: 25 April 2019
Published: 28 May 2019

Research article

Reducing file size and time complexity in secret sharing based document protection

Yan-Xiao Liu [1*], **Ya-Ze Zhang** [1], **Ching-Nung Yang** [2]

[1] Department of Computer Science and Engineering, XI'AN University of Technology, XI'AN, China

[2] Department of CSIE, National Dong Hwa University, Hualien, Taiwan

* **Correspondence:** Email: liuyanxiao@xaut.edu.cn.

Abstract: Recently, Tu and Hsu proposed a secret sharing based document protecting scheme. In their scheme, a document is encrypted into n shares using Shamir's (k, n) secret sharing, where the n shares are tied in with a cover document. The document reconstruction can be accomplished by acknowledgement of any k shares and the cover document. In this work, we construct a new document protecting scheme which is extended from Tu-Hsu's work. In Tu-Hsu's approach, each inner code of secret document takes one byte length, and shares are generated from all inner codes with the computation in $GF(257)$, where 257 is a Fermat Prime that satisfies $257 = 2^{2^3} + 1$. However, the share size expands when it equals to 255 or 256. In our scheme, each two inner codes of document is combined into one double-bytes inner code, and shares are generated from these combined inner codes with the computation in $GF(65537)$ instead, where 65537 is also a Fermat Prime that satisfies $65537 = 2^{2^4} + 1$. Using this approach, the share size in our scheme can be reduced from Tu-Hsu's scheme. In addition, since the number of combined inner codes is half of the inner codes number in Tu-Hsu's scheme, our scheme is capable of saving almost half running time for share generation and document reconstruction from Tu-Hsu's scheme.

Keywords: document protection; secret sharing; share size; Fermat Prime

1. Introduction

The rapid development of network technology has brought us into the era of informationization. The Internet facilitates the exchange of information with others, various web applications [1,2] greatly facilitate people's daily life. However, information security has become a major issue in the communication via Internet. Many approaches can be employed to protect secret information. (k, n) Secret sharing scheme is an important issue in cryptography which provides an efficient way to safely keeping secret key. In (k, n) Secret sharing, a secret is encrypted into n shares in such a way that any group of at least k shares can recover the secret and less than k shares get nothing on this secret. There are different approaches to achieve secret sharing, for instance, Shamir's scheme [3] was based on polynomial; Blakley's scheme [4] was based on geometry, and Chinese Reminder Theorem was another approach for secret sharing schemes [5,6].

Many kinds of digital information can be regarded as secret key that can be protected using Shamir's secret sharing scheme. For instance, (k, n) secret image sharing schemes [7–9] takes digital image as secret key, and encrypts it into meaningless shadows such that k or more shadows can reconstruct the image, less than k shadows get nothing on the secret image. In 2014, Tu and Hsu proposed a novel document protecting scheme [10] which is based on Shamir's (k, n) secret sharing. In their scheme, a secret document is regarded as secret key and is encrypted into n shares via a cover document, both at least k shares and cover document are necessary to recover the secret document, less than k shares or lack of cover image leads no information on the secret document. Comparing to other secret sharing based document protecting scheme [11,12], Tu-Hsu's scheme has the following advantages. First, the cover document looks innocent that would probably be ignored by hackers. Second, those schemes [11,13] adopts a (n, n) secret sharing scheme, on the contrary, Tu-Hsu's scheme uses (k, n) secret sharing which is more applicable than the schemes [11,13].

As we know, the size of share is an important issue in secret sharing, since smaller share size can reduce storage and computation cost. Lots of works [14–16] addressed the topic of reducing share size in secret sharing based cryptographic schemes. The share in Tu-Hsu's scheme is generated byte-wisely in $GF(257)$ from all inner codes of secret document, where 257 is a Fermat Prime that satisfies $257 = 2^{2^3} + 1$. The computation in $GF(257)$ can guarantee that all inner codes can be correctly recovered since it is the smallest prime larger than $2^8 = 256$. However, the computation in $GF(257)$ would cause share size expansion when the it equals to 255 or 256. In fact, the problem of share size expansion can be solved by using $GF(2^8)$ [17,18] instead of $GF(257)$, but the time complexity of computation in $GF(2^8)$ is much higher than running time in $GF(257)$. It needs to transform integers in $[0, 255]$ into corresponding polynomials in $GF(2^8)$, and then the computation between integers is transformed int computation between polynomials $(\text{mod} x^8 + x^4 + x^3 + x + 1)$. Some secret image sharing schemes computed shadows in $GF(251)$ that can resolve the problem of share size expansion, but it would cause image distortion during reconstruction. Therefore the approach of $GF(251)$ can not be adopted in document protecting scheme since each inner code of document should be correctly reconstructed.

In this paper, we construct a new secret sharing based document protecting scheme which is extended from Tu-Hsus scheme [10]. In their scheme, all inner codes of secret document are encrypted into shares single byte-wisely in $GF(257)$. Another reasonable choice is using $GF(2^8)$ rather than the ordinary arithmetic $GF(257)$, i.e., mod 251, to deal with sharing byte-wisely, so that the calculation can process the whole range $[0, 255]$. And, file size is not expanded, because we do not have the values of 255 and 256. However, mathematical calculations in polynomial processing under $GF(2^8)$ are complicated. If we process not byte-wisely, but deal with N bits each time. Tu-Hsus approach is based on $GF(257)$, i.e., $GF(2^8 + 1)$. Thus, we may use $GF(2^N + 1)$, where $2^N + 1$ is prime. In fact, the value of 257 in Tu-Hsus approach is a Fermat prime. As we know, the only known Fermat primes are $3, 5, 17, 257$, and 65537, where N is $1, 2, 4, 8$, and 16, respectively. Our motivation is still using a simple modular arithmetic, modularizing a Fermat prime. Different from their approach, our scheme combines each two inner codes of secret document into a new double-bytes inner code (i.e., $N = 16$), and all these double-byte inner codes are encrypted into shares with computation in $GF(65537)$, where 65537 is also a Fermat prime that equals to $2^{2^4} + 1$. Using our approach, each share takes two bytes storage space which can be read and stored efficiently in programs and the share size can be reduced from Tu-Hsus scheme. On the other hand, the number of combined inner codes in secret document is half of inner codes number using Tu-Hsus approach, thus the time complexity in our scheme is expected half of Tu-Hsus approach, experimental results demonstrate that our scheme is capable of saving almost half running time from Tu-Hsus scheme.

The rest of this paper is organized as follows. In next section, we introduce Shamir's (k, n) secret sharing scheme, Tu-Hsu's secret sharing based document protecting scheme and definition of Fermat Prime respectively. Our proposed scheme is described in Section 3, and the analysis on share size and running time in our scheme is also discussed in this part. In section 4, the comparisons of share size and running time between our scheme and Tu-Hsu's scheme are shown in the experimental results. The conclusion is made in section 5.

2. Preliminaries

2.1. *Shamir's (k, n) secret sharing scheme*

A (k, n) secret sharing scheme is a method where a secret is encrypted into n shares, in such way that any k or more shares can reconstruct the secret and fewer than k shares get nothing on the secret. More formally, in secret sharing scheme, there exists n users $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ and a dealer $\mathcal{D}$. In 1979, Shamir introduced a polynomial based (k, n) secret sharing scheme which is shown in following **Scheme 1**.

Scheme 1: *Shamir's (k, n) secret sharing scheme*

Sharing phase:

1 $\mathcal{D}$ randomly chooses a $k - 1$ degree polynomial $f(x) \in GF(q)[x]$ which satisfies $s = f(0) \in GF(q)$. (q is a prime number which satisfies the security requirement)
2 $\mathcal{D}$ selects n different integers $x_1, x_2, ..., x_n$ in $GF(q)$ as n different IDs and computes n shares $v_i = f(x_i), i = 1, 2, ..., n$.
3 $\mathcal{D}$ sends each share and its ID $(v_i, x_i), i \in [1, n]$ to P_i respectively.

Reconstruction phase:

1 $m(\geq k)$ users (say $P_1, P_2, ..., P_m$) pool their shares and IDs $(v_i, x_i), i = 1, 2, ..., m$ together.
2 Computing the interpolated polynomial $f(x)$ on $(v_i, x_i), i = 1, 2, ..., m$ by the following Lagrange equation:

$$f(x) = \sum_{i=1}^{m}(v_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}) \tag{2.1}$$

Then the secret $s = f(0)$.

2.2. Fermat Prime Number

The Fermat Number is a sequence of integers F_t that satisfies:

$$F_t = 2^{2^t} + 1, t \geq 0 \tag{2.2}$$

A prime number N is called Fermat Prime Number, when there exist $t \geq 0$, and satisfies that $N = 2^{2^t} + 1$. $F_0 = 1, F_1 = 3, F_2 = 17, F_3 = 257$ and $F_4 = 65537$ are five Fermat Prime Numbers and any Fermat Prime Number F_t for $t \geq 5$ has not been found yet.

2.3. Tu-Hsu's scheme

Tu and Hsu constructed a document protecting scheme using Shamir's (k, n) secret sharing. Their scheme can be also divided into **Sharing Phase** and **Reconstruction Phase**: During **Sharing Phase**, a secret document **SD** was encrypted into n shares on different IDs, where these IDs are generated from a cover document **CD**; in **Reconstruction Phase**, acknowledgement of **CD** and a group of at least k shares can reconstruct **SD**. Before **Sharing Phase**, all the words in **SD** and **CD** are encoded into inner codes using an appropriate encoding method. We use the notations $S_i, i = 1, 2, ..., m$ and $C_i, i = 1, 2, ..., l$ to present the inner codes of **SD** and **CD** in byte-wise respectively. Tu-Hsu's scheme is described in following **Scheme 2**.

Scheme 2: *Tu-Hsu's (k, n) secret sharing based document protecting scheme*

Sharing phase: Input: secret document **SD** $= (S_1, S_2, ..., S_m)$, cover document **CD** $= (C_1, C_2, ..., C_l)$; Output: n shares $V_1, V_2, ..., V_n$

1 The l inner codes $(C_1, C_2, ..., C_l)$ is divided into multiple n-length blocks $B_1, B_2, ..., B_{\lfloor \frac{l}{n} \rfloor}$, where the n inner codes in each block are different. The method for generating these n-length blocks is introduced in following **Algorithm 1**.

2 For each $k-1$ inner codes $(S_{r(k-1)+1}, S_{r(k-1)+2}, ..., S_{r(k-1)+k-1}, r = 0, 1, ..., \frac{m}{k-1})$ in SD, $\mathcal{D}$ constructs a $k-1$ degree polynomial $f_r(x) \in GF(257)[X]$ using these $k-1$ inner codes as coefficients, and the constant term $A_{r,0}$ is randomly selected:

$$f_r(x) = A_{r,0} + S_{r(k-1)+1} \cdot x + ... + S_{r(k-1)+k-1} \cdot x^{k-1} (\text{mod } 257) \tag{2.3}$$

(The reason for only $k-1$ (not k) inner codes are used as coefficients is to enhance the security level.)

3 For each polynomial $f_r(x), r \in [0, \frac{m}{k-1}]$, using n different inner codes $x_{r,j}, j = 1, 2, ..., n$ in $B_{(r+1) mod \lfloor \frac{l}{n} \rfloor}$ as different IDs to compute n sub-shares $v_{r,j} = f_r(x_{r,j}), j = 1, 2, ..., n$.

4 The share $V_j, j = 1, 2, ..., n$ for each user P_j is

$$V_j = v_{1,j} || v_{2,j} || ... || v_{\frac{m}{k-1}, j} \tag{2.4}$$

Reconstruction phase: Input cover document **CD**, k shares $V_1, V_2, ..., V_k$; Output: secret document **SD**

1 Obtain the multiple n-length blocks $B_1, B_2, ..., B_{\lfloor \frac{l}{n} \rfloor}$ from **CD** using following **Algorithm 1**.

2 Reconstructing $f_r(x), r = 0, 1, ..., \frac{m}{k-1}$ using Lagrange interpolation:

$$f_r(x) = \sum_{i=1}^{k}(v_{r,i} \prod_{j=1,j\neq i}^{k} \frac{x - x_{r,j}}{x_{r,i} - x_{r,j}}) \tag{2.5}$$

where $x_{r,i}, i \in [1, k]$ are the IDs of $v_{r,i}$ that are obtained from $B_{(r+1)mod\lfloor \frac{l}{2n} \rfloor}$

3 The last $k - 1$ coefficients in $f_r(x)$ are $k - 1$ corresponding inner codes of **SD**, thus **SD** (all inner codes) can be recovered.

The method of dividing all inner codes in **CD** into multiple n-length blocks is described in following **Algorithm 1**. Using **Algorithm 1**, one can guarantee that the n elements in each block are different.

Algorithm 1 Dividing CD into multiple n-length blocks

1 Sort all inner codes $(C_0, C_1, ..., C_{l-1})$ of CD in ascending order, and $(C'_0, C'_1, ..., C'_{l-1})$ is sorted inner codes.
2 Put C'_1 into B_0 and set $count = 1, m = 0$
3 For ($j = 1$ to $j = l - 1$)
{if ($C'_j > C'_{j-1}$ or $count < \lfloor \frac{l-1}{n} \rfloor$)
{if ($C'_j == C'_{j-1}$)
$count = count + 1$
else
$count = 1$}
$m = (m + 1)\%\lfloor \frac{l-1}{n} \rfloor$
if ($|B_m| < n$) /* $|B_m|$ denotes the size of B_m */
put C'_j into B_m }

In **Sharing Phase**, Tu and Hsu selects the prime 257 in the finite field $GF(257)$ to computing shares and reconstructing document. As introduced previously, 257 is a Fermat Prime that satisfies $257 = 2^8 + 1$, it is only 1 larger than $2^8 = 256$, where 256 is exactly one byte length. Since each inner code of secret document in Tu-Hsu's scheme takes one byte storage space, the share generation and secret reconstruction achieve highest efficiency when the the corresponding computation is in $GF(257)$.

However, computation in $GF(257)$ would causes some sub-shares equal to 256, that cannot be stored in one byte space. To ensure the correctness of each sub-share, they adopt a special way to present 255 and 256. The sub-share 255 is stored in two bytes 255||0, and sub-share 256 is stored in two bytes 255||1. During **Reconstruction phase**, when a sub-share is 255, one should know that the next byte is also combined with previous byte. If the value is 0 in next byte, the sub-share is 255, otherwise the sub-share is 256. For instance, if a group of sub-shares is $(45, 79, 255, 0, 80, 178)$ which is stored in 6 bytes, then there are 5 sub-shares $(45, 79, 255, 80, 178)$ in total; if a group of sub-shares is $(97, 255, 1, 75)$ which is stored in 4 bytes, there are 3 sub-shares $97, 256, 75$; if a group of sub-shares is $(189, 253, 94, 180)$ which is stored in 4 bytes, there are 4 sub-shares $189, 253, 94, 180$.

3. Proposed scheme

The share size is an important issue in secret sharing scheme. For instance, Shamir's (k, n) secret sharing hides the secret s in one coefficient a_0, thus the size of share equals to the size of secret, $|V| = |S|$; (k, n) secret image sharing scheme uses all k coefficients to hide a group of k pixels, the share size is $\frac{1}{k}$ time of secret image, $|V| = \frac{|S|}{k}$. Tu-Hsu's scheme uses $k-1$ out of k coefficients to hide a group of $k-1$ inner codes of secret document, the theoretical share size is $|V| = \frac{|S|}{k-1}$. However, Tu-Hsu's scheme uses $GF(257)$ in share generation, when the share belongs to $\{0, 1, ..., 254\}$, it can be stored in one byte, and there is no share size expansion; but when the share equals to 255 or 256, it is stored in two bytes, this would causes share size expansion from the theoretical size.

In this paper, we aim to construct a new secret sharing based secret document protecting scheme that can reduce share size from Tu-Hsu's scheme. In Tu-Hsu's scheme, all inner codes of secret document are encrypted single-byte wisely in $GF(257)$, and it causes share size expansion when the share equals to 255 or 256. The probability of share size expansion is $\frac{2}{257}$. Different from their approach, our scheme first combines each two inner codes of secret document into one inner code, where each new inner code in our approach takes double-bytes storage space. Then the shares are generated from this double-bytes inner codes with the computation in $GF(65537)$, and the secret reconstruction is also computed in $GF(65537)$ from double-byte shares. Notice that 65537 is also a Fermat Prime Number which is introduced previously, and 65537 is the only Fermat Prime Number which is larger than 257. The computation in $GF(65537)$ has following advantages:

1 65537 is only 1 larger than $2^{16} = 65536$, where 63336 is exact two bytes length. Therefore the computation in $GF(65537)$ has high efficiency as the computation in $GF(257)$.

2 The share size using our approach would expand when the share equals to 65535 or 65536. The probability of share size expansion is $\frac{2}{65537}$, which is much smaller than the probability $\frac{2}{257}$ of share size expansion in Tu-Hsu's approach. Therefore our scheme can reduce share size from Tu-Hsu's scheme.

3 The number of inner codes in our approach is half inner codes number in Tu-Hsu's approach, thus our scheme has less time complexity than Tu-Hsu's scheme.

Our proposed scheme is described in following **Scheme 3**.

Scheme 3: *Our proposed scheme*

Sharing phase: Input: secret document $SD = (S_1, S_2, ..., S_m)$, cover document $CD = (C_1, C_2, ..., C_l)$; Output: n shares $V_1, V_2, ..., V_n$

1 Combine each of two inner codes in SD and CD, thus $SD = (S^*_1, S^*_2, ..., S^*_{\frac{m}{2}})$ and $CD = (C^*_1, C^*_2, ..., C^*_{\frac{l}{2}})$. Each new inner code in SD and CD is stored in double-bytes.

2 Dividing $(C^*_1, C^*_2, ..., C^*_{\frac{l}{2}})$ into multiple n-length blocks $B^*_1, B^*_2, ..., B^*_{\lfloor \frac{l}{2n} \rfloor}$ using **Algorithm 1**.

3 For each group of $k-1$ inner codes $(S^*_{r(k-1)+1}, S^*_{r(k-1)+2}, ..., S^*_{r(k-1)+k-1}, r \in [0, \frac{m}{2(k-1)}]$ in SD, $\mathcal{D}$ randomly selects an integer $A_{r,0} \in [0, 65536]$ and generates a $k-1$ degree polynomial $f^*_r(x)$:

$$f^*_r(x) = A_{r,0} + S^*_{r(k-1)+1} \cdot x + ... + S^*_{r(k-1)+k-1} \cdot x^{k-1} (\text{mod } 65537) \tag{3.1}$$

4 For each polynomial $f^*_r(x), r \in [0, \frac{m}{2(k-1)}]$, using n different integers $x_{r,j}, j = 1, 2, ..., n$ in $B_{(r+1) mod w}$ as n different IDs to compute n sub-shares $v_{r,j} = f_r(x_{r,j}), j = 1, 2, ..., n$. If a sub-

share is 65535 or 65536, it is stored in three bytes where the first double-bytes is set 65535 and the last byte is set 0 or 1 respectively.

5 The share $V_j, j = 1, 2, ..., n$ for each user P_j is

$$V_j = v_{0,j}||v_{1,j}||...||v_{\frac{m}{2k-1},j} \tag{3.2}$$

Reconstruction phase: Input cover document CD, k shares $V_1, V_2, ..., V_k$; Output: secret document SD

1 Obtain the n-length blocks $B^*_1, B^*_2, ..., B^*_{\lfloor \frac{l}{2n} \rfloor}$ from CD using **Algorithm 1**.

2 Reconstructing the polynomials $f^*_r(x), r = 0, 1, ..., \frac{m}{2(k-1)}$ using Lagrange interpolation:

$$f^*_r(x) = \sum_{i=1}^{k}(v_{r,i} \prod_{j \neq i} \frac{x - x_{r,j}}{x_{r,i} - x_{r,j}}) \tag{3.3}$$

where $x_{r,i}, i \in [1, k]$ are the IDs of $v_{r,i}$ that are obtained from $B^*_{(r+1) \bmod \lfloor \frac{l}{2n} \rfloor}$

3 The last $k - 1$ coefficients in $f^*_r(x)$ are $k - 1$ inner codes of SD, thus SD (all inner codes) can be recovered correspondingly.

The difference between our scheme and Tu-Hsu's scheme is that the proposed scheme combines each two bytes of inner codes into a double-bytes block, and computing sub-shares in $GF(P)$ where $P = 65537$. In Tu-Hsu's scheme, sub-shares are computed in $GF(257)$, and the sub-share size expands from one byte to two bytes when the sub-share equals to 255 or 256, the probability that the sub-share equals to 255 or 256 is $\frac{2}{257}$, thus the theoretical average share size (combined by all sub-shares) is

$$|V_{Tu-Hsu}| = \frac{|S|}{257} \cdot (\frac{255}{k-1} + \frac{2}{k-1} \cdot 2) = \frac{259|S|}{257(k-1)} \tag{3.4}$$

In our scheme, the sub-share size expands from double-bytes to three bytes when sub-share equals to 63355 or 65536, the probability that the sub-share equals to 65535 or 65536 is $\frac{2}{65537}$, thus the theoretical average share size (combined by all sub-shares) is

$$|V_{Pro}| = \frac{|S|}{63357} \cdot (\frac{65535}{k-1} + \frac{2}{k-1} \cdot \frac{3}{2}) = \frac{65538|S|}{65537(k-1)} \tag{3.5}$$

It is obvious that $|V_{Pro}| < |V_{Tu-Hsu}|$, therefore our scheme can reduce share size of Tu-Hsu's scheme.

In addition, since our scheme encrypts secret document double-bytes wisely, the number of inner codes for a secret document in our scheme is $\frac{1}{2}$ times of Tu-Hsu's scheme. As analyzed in [10], the time complexities for share generation and secret document reconstruction is $O(hn)$ and $O(hk)$ respectively, where h denotes the number of inner codes of secret document. Since the multiplications in $GF(257)$ and $GF(65537)$ have similar running time, the time complexities in our scheme are $O(\frac{hn}{2})$ and $O(\frac{hk}{2})$ for share generation and secret document reconstruction respectively. It means that our approach is capable of saving half running time from Tu-Hsu's approach.

4. Comparisons

Cover Document:

中颱納坦再現「雙眼牆」情況，「雙眼牆」現象又可稱為「雙眼皮」，即有雙颱風眼，並有一大一小同心圓「雙眼牆」通常只會出現在強烈颱風，當其結構強度發展到最高極限時，就會在颱風眼內部再長出一個小颱風眼，出現兩圈眼牆，小颱風眼會繞著大颱風眼繞圈圈，一直到小颱風眼結構減弱被大颱風眼「吃掉」為止.

Secret Document:

法務部密令:8月1日起執行代號“Anti-Pirate”的反盜版行動，請各檢警單位配合

Figure 1. Content of secret document and cover document.

Cover document:

164 164 187 228 175 199 169 90 166 65 178 123 161 117 194 249 178 180 192 240 161 118 177 161 170 112 161 65 161 117 194 249 178 180 192 240 161 118 178 123 182 72 164 83 165 105 186 217 172 176 161 117 194 249 178 180 165 214 161 118 161 65 167 89 166 179 194 249 187 228 173 183 178 180 161 65 168 195 166 179 164 64 164 106 164 64 164 112 166 80 164 223 182 234 161 67 161 117 194 249 178 180 192 240 161118 179 113 177 96 165 117 183 124 165 88 178 123 166 98 177 106 175 80 187 228 173 183 161 65 183 237 168 228 181 178 186 99 177 106 171 215 181 111 174 105 168 236 179 204 176 170 183 165 173 173 174 201 161 65 180 78 183 124 166 98 187 228 173 183 178 180 164 186 179 161 166 65 170 248 165 88 164 64 173 211 164 112 187 228 173 183 178 180 161 65 165 88 178 123 168 226 176 233 178 180 192 240 161 65 164 112 187 228 173 183 178 180 183 124 194 182 181 219 164 106 187 228 173 183 178 180 194 182 176 233 176 233 161 65 164 64 170 189 168 236 164 112 187 228 173 183 178 180 181 178 186 99 180 238 174 122 179 81 164 106 187 228 173 183 178 180 161 117 166 89 177 188 161 118 172 176 164 238 161 67

Secret document:

170 107 176 200 179 161 177 75 165 79 161 71 56 164 235 49 164 233 176 95 176 245 166 230 165 78 184 185 161 167 65 110 116 105 45 80 105 114 97 116 101 161 168 170 186 164 207 181 115 170 169 166 230 176 202 161 65 189 208 166 85 192 203 196 181 179 230 166 236 176 116 166 88 161 67 32

Figure 2. Inner codes of secret document and cover document.

In this section, we use experimental results to show the advantages of our scheme to Tu-Hsu' scheme. Our experiments adopt the same samples in [10] as the secret document and cover document, and three different thresholds $(2, 5), (3, 6), (4, 7)$ secret sharing schemes were implemented on Tu-Hsu's approach and our approach using Matlab language, respectively. The program runs at platform of CPU i5-7300HQ, and 8.0 GB RAM, and the operating system is Window 7 Professional. The share size and running time are compared in three thresholds secret sharing schemes between these two approaches. Figure 1 shows the secret document and cover document, both are in traditional Chinese. Figure 2 lists the inner codes of the secret document (76 bytes) and cover document, which are transformed by the encoding method Big5.

Experiment 1: $(2, 5)$ secret sharing based on secret document and cover document using Tu-Hsu's approach and our approach.

In Experiment 1, the secret document is first encoded into 5 shares using Tu-Hsu's $(2, 5)$ secret sharing where the all inner codes of secret document is encrypted single byte-wisely. Then we use our approach to encode secret document into 5 shares where the inner codes are encrypted double-byte wisely. Figure 3 lists the shares that are generated in Tu-Hsu's approach and our approach respectively.

$(2, 5)$ secret sharing scheme uses only 1 inner code as a coefficient in polynomials to generating shares, thus the size of share would equals to the size of secret document theoretically. From Figure 3 we can see that the sizes of 5 share using Tu-Hsu's approach are $76, 76, 77, 78, 76$ bytes respectively. The share size expansion are caused by the three sub-shares $256, 256, 255$ (marked in red), which are stored in two bytes. On the other hand, the sizes of 5 shares using our approach are all 76 bytes, there is no share size expansion using our approach.

Experiment 2: $(3, 6)$ secret sharing based on secret document and cover document using Tu-Hsu's approach and our approach.

In Experiment 2, secret document is first encoded into 6 shares using Tu-Hsu's $(3, 6)$ secret sharing where the all inner codes of secret image is encrypted byte-wisely. Then we use our approach to encode secret document into 6 shares where the inner codes are encrypted double-byte wisely. Figure 4 lists a group of 6 shares that are generated in Tu-Hsu's approach and our approach respectively.

Since both $(3, 6)$ secret sharing in Tu-Hsu's approach and our approach takes a group of 2 inner codes as coefficients in a polynomial, the theoretical share size is $\frac{1}{2}$ of the secret document. As listed in Figure 4, share 1 and 5 consists of 39 bytes which is caused by the sub-shares marked in red, and each share in our approach is 38 bytes which equals to $\frac{1}{2}$ of secret document.

Experiment 3: $(4, 7)$ secret sharing based on secret document and cover document using Tu-Hsu's approach and our approach.

In Experiment 3, secret document is first encoded into 7 shares using Tu-Hsu's $(4, 7)$ secret sharing where the all inner codes of secret image is encrypted byte-wisely. Then we use our approach to encode secret document into 7 shares where the inner codes are encrypted double-byte wisely. Figure 5 lists a group of 7 shares that are generated in Tu-Hsu's approach and our approach respectively. We can also observe that the share size in our approach is smaller than the share size in Tu-Hsu' approach.

Shares using Tu-Hsu's approach in (2,5) secret sharing

Share1:

168	81	48	77	223	82	194	88	163	149	134	69	137	33	191	155	156	161	17	96
229	205	128	176	119	166	232	249	204	30	111	56	127	52	212	201	221	234	228	152
180	54	125	120	31	66	208	173	136	112	20	109	253	166	154	1	5	150	119	245
68	105	8	68	77	70	228	183	106	53	1	187	125	8	92	28				

Share2:

5	226	152	55	68	72	100	144	69	23	170	203	116	100	179	135	103	225	138	81
174	40	167	4	85	128	116	20	132	14	32	10	214	79	40	215	143	164	200	198
136	202	123	222	194	113	228	123	2	48	165	110	242	214	48	248	242	250	104	16
200	125	221	123	234	35	206	185	188	149	45	87	57	199	55	211				

Share3:

32	201	10	174	234	217	178	103	196	62	173	71	106	34	238	111	108	218	194	193
230	153	103	194	9	36	11	184	8	129	168	82	205	206	66	147	118	247	79	48
164	239	251	119	42	96	42	163	212	202	49	212	148	189	198	232	210	153	4	124
85	92	100	210	140	104	220	137	255	1	110	36	114	85	75	88	81			

Share4:

16	200	66	4	147	189	69	157	212	160	241	10	152	132	18	38	135	26	141	87
177	69	237	5	136	146	198	122	11	180	174	191	246	18	124	193	82	17	244	89
255	1	177	231	111	130	145	156	69	104	20	28	164	162	231	52	108	219	108	236
151	249	148	20	53	151	95	207	255	0	46	152	193	157	187	174	43	48		

Share5:

4	242	203	234	83	226	57	254	117	60	86	233	165	115	202	198	238	221	200	233
169	139	29	129	16	214	108	79	125	152	80	111	185	117	93	81	192	63	51	69
212	133	219	209	195	241	26	77	38	193	226	176	30	92	71	176	163	184	64	253
69	201	238	166	39	31	86	9	245	155	59	29	88	251	75	204				

Shares using our approach in (2,5) secret sharing

Share1:

58271	34141	56400	44752	59708	41713	49280	28684	35829	19953
61022	15121	53277	48281	1782	25234	49345	3452	58243	34520
51646	41845	21265	49781	28745	26942	40829	20255	32114	39357
15025	52308	62590	36355	47405	9555	1494	10408		

Share2:

32694	17228	16961	35568	23804	52777	52799	42017	25367	29007
18939	53523	52511	435	25101	26846	14617	48734	16499	23661
35822	1805	35744	64727	27554	8126	32956	33166	11870	62052
22043	35044	5826	46608	17798	37589	6716	24025		

Share3:

51709	62663	28206	21099	50804	34369	35429	4196	58738	17379
53662	44835	40151	32814	4326	63272	14743	45260	12742	60889
20341	61182	19271	47433	55189	4990	20721	272	17207	35527
44952	32388	58732	40744	23995	2711	16836	33622		

Share4:

50937	15902	55730	41447	57554	29767	63855	11125	64366	55281
47913	13336	4572	46945	24934	17411	56540	6986	5089	38914
5849	35743	26046	26422	311	24745	34410	22034	20765	4080
23957	56478	56652	54353	54917	15134	44585	22836		

Share5:

50358	9703	37269	21495	60187	42202	55744	62279	42251	11639
6521	21667	7191	53621	55846	7976	26991	23238	55459	35165
64463	60441	54430	19066	51631	34308	57488	35196	25284	36255
50694	31153	33621	24406	29003	883	64906	30055		

Figure 3. Shares in Experiment 1.

Shares using Tu-Hsu's approach in (3,6) secret sharing

Share1:

255 0 169 41 21 36 63 107 73 62 22 182 157 216 77 48 133 217 109 186

108 61 223 66 86 130 62 222 0 144 105 153 233 10 186 229 79 51 185

Share2:

6 83 179 25 178 241 54 250 82 158 12 133 119 2 51 190 93 239 241 7

238 30 108 214 223 12 38 240 0 201 145 188 65 98 120 99 150 139

Share3:

192 85 20 167 156 118 22 76 72 132 67 221 151 81 143 171 150 186 60 229

39 232 99 92 88 49 145 44 107 254 130 196 118 158 212 86 18 129

Share4:

120 136 122 236 241 139 72 43 238 155 167 229 222 38 225 71 12 207 60 165

229 85 72 75 148 162 2 149 118 243 159 74 37 0 186 130 174 66

Share5:

226 143 65 230 198 11 219 18 109 219 187 26 221 218 105 79 63 141 241 61

158 146 187 140 201 204 152 69 255 0 83 197 111 123 227 164 91 177 236

Share6:

28 219 37 142 69 86 158 168 186 219 43 121 58 137 193 102 1 161 85 41

170 8 142 116 248 147 180 236 72 91 208 194 78 4 13 146 60 155

Shares using our approach in (3,6) secret sharing

Share1:

27955 42244 3885 11295 29765 25491 65165 59420 3627 28408

61426 48390 22064 49409 29376 5150 54943 46089 32875

Share2:

4574 24243 44634 707 4615 10598 42309 16488 57100 25324

4995 59629 25840 40078 42378 42574 19415 9276 609

Share3:

21948 47524 11549 46679 10798 58060 29415 45378 53917 2819

53485 35232 8945 34594 60287 36877 49714 52544 28984

Share4:

48339 34558 63304 9103 19952 36882 11127 2587 28593 30717

30738 59348 12660 44975 55415 9830 57378 53980 63622

Share5:

47497 45381 60152 51795 25700 13580 16825 26769 42832 1633

30967 2956 62993 62065 5985 26605 13141 58751 64112

Share6:

47066 42971 34954 14001 45582 27979 47704 62080 1458 21975

17637 46924 41264 57825 3402 64802 55413 50228 155

Figure 4. Shares in Experiment 2.

As we discussed previously, the operations in $GF(257)$ and $GF(65537)$ have similar time complexity, and thus the share encryption and secret reconstruction using our scheme is expected to save $\frac{1}{2}$ running time from Tu-Hsu's approach. To authenticate this assumption, we implement the all previous experiments and a $(5, 8)$ secret sharing based secret document protection scheme three times, and record the running times of Tu-Hsu's approach and our approach respectively, the following Table 1 lists all the data from these experiments which includes the total share size, running time for share generation and secret reconstruction.

Shares using Tu-Hsu's approach in (4,7) secret sharing

Share1:
36 60 197 53 187 81 230 21 202 199 66 94 11 152 29 218 157 107 228 106
65 55 146 232 145 234
Share2:
24 23 52 11 35 87 77 102 247 12 239 238 113 67 27 54 171 58 46 94
127 180 182 171 162 60
Share3:
255 0 84 0 41 122 183 72 18 79 29 130 144 168 193 205 83 107 195 15
116 31 232 74 64 180 185
Share4:
97 126 132 30 47 156 159 5 44 150 130 131 78 187 184 255 1 22 45 25
132 41 193 168 219 72 107
Share5:
123 165 35 175 65 206 18 127 158 208 118 250 149 191 249 10 25 154 179 84
69 52 78 224 179 29
Share6:
138 13 157 39 240 247 157 112 22 214 163 122 120 223 8 94 251 92 114 73
24 90 233 209 145 211
Share7:
48 65 212 113 190 55 145 218 39 34 41 37 36 54 69 55 122 232 36 196
192 146 126 37 179 65

Shares using our approach in (4,7) secret sharing

Share1:
18929 44706 35619 48367 60693 17610 35404 60825 28290 20727
44620 60006 61633
Share2:
8924 6662 4339 36495 61370 61394 51442 43076 48786 10033
64906 5925 20595
Share3:
47539 64113 41866 53985 6965 40093 6386 56998 17346 9834
25191 27354 5163
Share4:
35412 46162 58276 20231 12496 15303 36430 37589 25031 25414
14125 36989 2655
Share5:
37867 32269 12668 58607 64203 51113 65308 54469 53121 30292
45203 23651 54158
Share6:
5583 27346 13852 30865 53588 36532 6180 58427 56207 63840
37205 17469 26696
Share7:
51030 4816 17943 24338 59530 43649 59583 64347 23464 56232
16618 9384 53433

Figure 5. Shares in Experiment 3.

Table 1. Comparisons between Tu-Hsu's approach and our approach.

Threshold	Approach	Total Share Size (byte)			Running Time (second)					
					Share Generation			Secret Reconstruction		
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
(2, 5)	Tu-Hsu's	383	382	383	0.007	0.007	0.011	0.448	0.423	0.417
	Our	380	380	380	0.004	0.005	0.006	0.256	0.239	0.217
(3, 6)	Tu-Hsu's	230	228	230	0.014	0.013	0.012	0.403	0.399	0.407
	Our	228	228	228	0.006	0.008	0.009	0.227	0.263	0.230
(4, 7)	Tu-Hsu's	184	184	183	0.013	0.011	0.010	0.479	0.469	0.455
	Our	182	182	182	0.009	0.007	0.005	0.252	0.246	0.258
(5, 8)	Tu-Hsu's	155	154	155	0.016	0.014	0.015	0.489	0.495	0.493
	Our	152	152	152	0.008	0.008	0.009	0.261	0.259	0.258

The statistical results in Table 1 shows that our approach is capable of reducing share size from Tu-Hsu's approach and also save almost half running time in share generation or secret reconstruction. Next, we use 10 secret documents with different sizes (100 bytes to 1000 bytes) to test the running times for secret reconstruction with three approaches: computations in $GF(2^8), GF(257)$ and $GF(65537)$ respectively. The following Table 2 lists all the running times for secret reconstruction with three thresholds $(2,5), (3,6), (4,7)$, and Figures 6–8 show the comparisons of running time between three computation approaches under different threshold respectively. From the comparison we can see that the computation in $GF(65537)$ is capable of saving half running time for secret reconstruction from computation in $GF(257)$ and also reducing the share size. Although the problem of share size expansion can be also solved by the computation in $GF(2^8)$, the running time in $GF(2^8)$ is much longer than the computations in $GF(257)$ and $GF(65537)$.

Table 2. Running times for secret reconstruction using three approaches.

Threshold			(2, 5)			(3, 6)			(4, 7)	
Approach		$GF(257)$	$GF(65537)$	$GF(2^8)$	$GF(257)$	$GF(65537)$	$GF(2^8)$	$GF(257)$	$GF(65537)$	$GF(2^8)$
Size	100	0.58	0.34	5.46	0.59	0.29	3.78	0.63	0.33	3.42
	200	1.15	0.64	11.01	1.15	0.62	8.01	1.27	0.66	7.05
	300	1.86	1.02	16.97	1.88	0.62	11.84	1.88	0.65	9.05
	400	2.13	1.18	17.71	2.06	1.09	13.20	2.29	1.14	12.08
	500	2.68	1.46	21.82	2.46	1.23	16.39	2.76	1.46	14.30
	600	3.47	1.82	27.11	3.27	1.66	22.30	3.58	1.82	20.01
	700	3.99	2.35	36.58	3.92	2.09	26.49	4.79	2.21	24.48
	800	4.58	2.54	42.72	4.43	2.26	30.44	4.89	2.52	26.90
	900	5.21	2.78	46.94	4.93	2.48	34.12	5.36	2.86	30.20
	1000	5.80	3.10	54.24	5.33	2.80	33.33	6.19	3.20	34.40

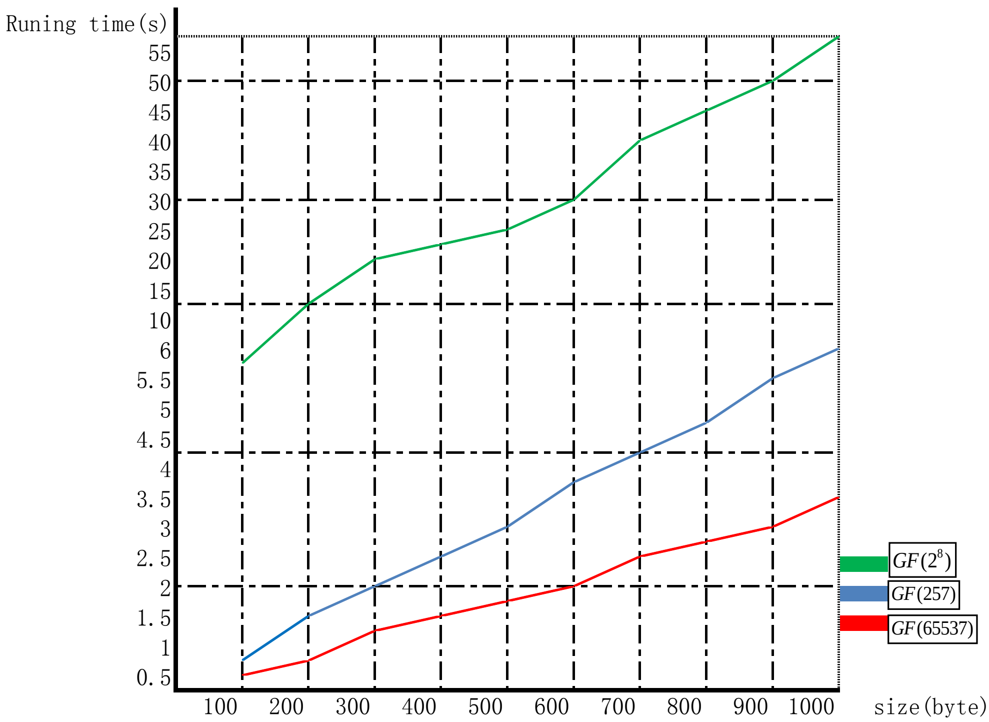

Figure 6. Running time for (2, 5) threshold secret reconstruction using three approaches.

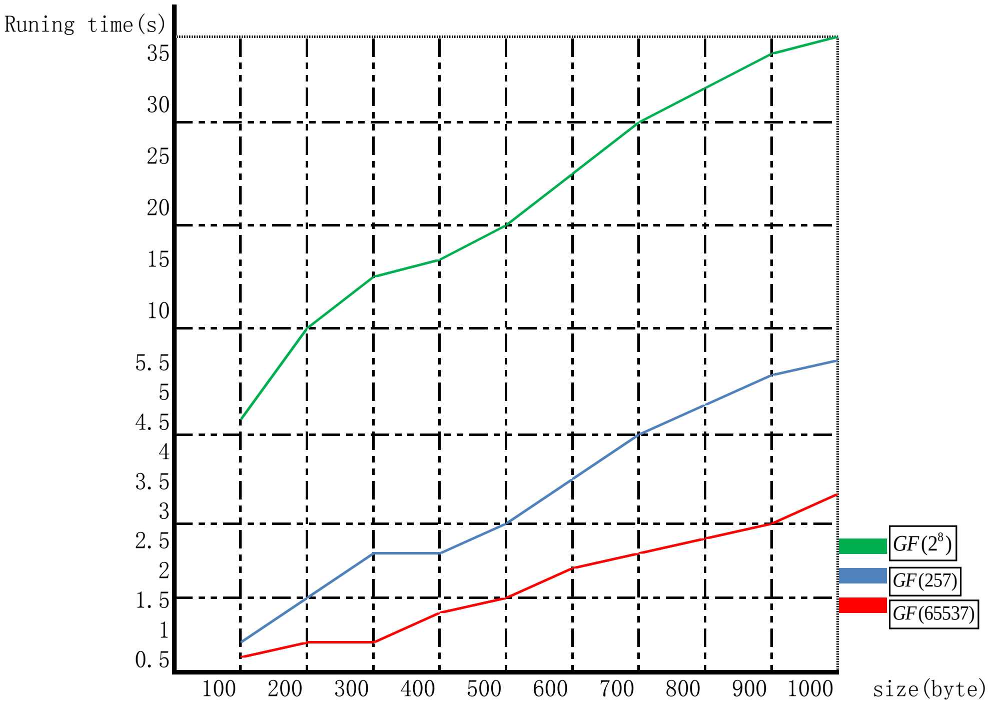

Figure 7. Running time for (3, 6) threshold secret reconstruction using three approaches.

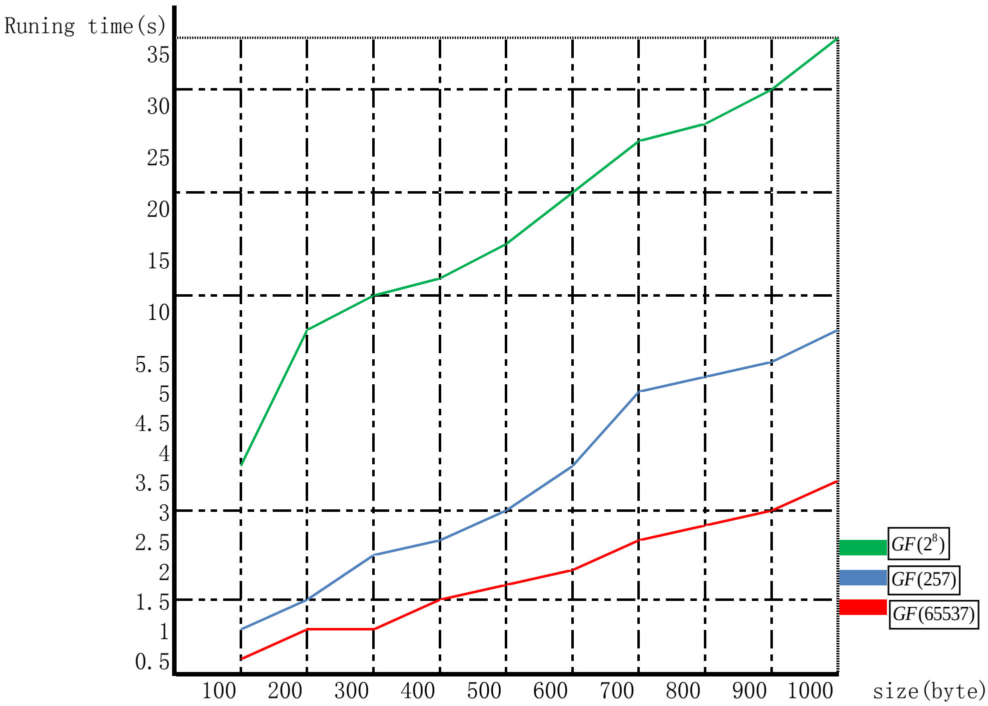

Figure 8. Running time for (4, 7) threshold secret reconstruction using three approaches.

5. Conclusion

This paper proposes a new secret sharing based secret document protection scheme, which is capable of reducing share size and saving running time from Tu-Hsu's scheme. In Tu-Hsu's scheme, all inner codes of a secret document are encrypted single-byte wisely, and the shares are computed through Shamir's (k, n) secret sharing in $GF(257)$. When the share in Tu-Hsu's scheme equals to 255 or 256, the share size is expanded from one byte to two bytes. On the contrary, our scheme combines each two inner codes of secret document in Tu-Hsu' scheme into one double-bytes inner code, and the shares are generated from these double-bytes inner codes through Shamir's (k, n) secret sharing in $GF(65537)$. The share size would expand only when it equals to 65535 or 65536, which has much smaller probability of share size expansion than Tu-Hsu's scheme. Thus, our scheme can reduce share size from Tu-Hsu's scheme. On the other hand, by combining each two inner codes into one double-bytes inner code, our scheme has half computational complexity to Tu-Hsu's scheme, and the experimental results can also prove that our scheme saves almost half running time from Tu-Hsu's scheme.

Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant No. 61502384,61571360,61872289; This research was supported in part by Ministry of Science and Technology (MOST), under Grant 107-2221-E-259-007.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Q. D. Sun, N. Wang, Y. D. Zhou, et al., Identification of influential online social network users based on Multi-Features, *Int. J. Pattern Recognit. Artif. Intell.*, **30** (2016), 1–15.
2. Q. D. Sun, N. Wang, S. C. Li, et al., Local spatial obesity analysis and estimation using online social network sensors, *J. Biomed. Inform.*, **83** (2018), 54–62.
3. A. Shamir, How to share a secret, *Commun. ACM*, **22** (1979), 612–613.
4. G. R. Blakley, Safeguarding cryptographic keys, *AFIPS Conference*, **48** (1979), 313–317.
5. X. X. Jia, Y. X. Song, D. S. Wang, et al., A collaborative secret sharing scheme based on the Chinese Remainder Theorem, *Math. Biosci. Eng.*, **16** (2019), 1280–1299.
6. X. X. Jia, D. S. Wang, D. X. Nie, et al., A new threshold changeable secret sharing scheme based on the Chinese Remainder Theorem, *Inf. Sci.*, **473** (2019), 13–30.
7. C. C. Thien and J. C. Lin, Secret image sharing, *Comput. Graph.*, **26** (2002), 765–770.
8. Y. X. Liu, C. N. Yang, C. M. Wu, et al. Threshold changeable secret image sharing scheme based on interpolation polynomial, *Multimed. Tools Appl.*, (2019), (DOI: 10.1007/s11042-019-7205-4).

9. Y. X. Liu and C. N. Yang, Scalable secret image sharing scheme with essential shadows. *Signal Process. Image*, **58** (2017), 49–55.
10. S. F. Tu and C. S. Hsu, Protecting secret documents via a sharing and hiding scheme, *Inf. Sci.*, **279** (2014), 52–59.
11. C. C. Chang and T. X. Yu, Sharing a secret gray image in multiple images, *First International Symposium on Cyber Worlds*, (2002), 230–237.
12. D. S. Tsai, G. Horng, T. H. Chen, et al., A novel secret image sharing scheme for true-color images with size constraint, *Inf. Sci.*, **179** (2009), 3247–3254.
13. R. Lukac and K.N. Plataniotis, Bit-level based secret sharing for image encryption, *Pattern Recognit.*, **38** (2005), 767–772.
14. Y. X. Liu, C. N. Yang and P. H. Yeh, Reducing shadow size in smooth scalable secret image sharing, *Secur. Commun. Netw.*, **7** (2014), 2237–2244.
15. R. Z. Wang and C. H. Su, Secret image sharing with smaller shadow images, *Pattern Recogn. Lett.*, **27** (2006), 551–555.
16. C. N. Yang, P. Li, C. C Wu, et al., Reducing shadow size in essential secret image sharing by conjunctive hierarchical approach, *Signal Process. Image*, **31** (2015), 1–9.
17. C. N. Yang, J. F. Ouyang and L. Harn et al., Steganography and authentication in image sharing without parity bits, *Opt. Commun.*, **285** (2012), 1725–1735.
18. C. C. Chen and S. C. Chen, Two-layered structure for optimally essential secret image sharing scheme, *J. Vis. Commun. Image R.*, **38** (2016), 595–601.