symmetry

MDPI

Article

A Novel Threshold Changeable Secret Image Sharing Scheme

Guohua Wu [1], Mingyao Wang [1], Qiuhua Wang [1], Ye Yao [1], Lifeng Yuan [1,2,*] and Gongxun Miao [3]

[1] School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China; wugh@hdu.edu.cn (G.W.); yao@hdu.edu.cn (M.W.); wangqiuhua@hdu.edu.cn (Q.W.); yaoye@hdu.edu.cn (Y.Y.)
[2] Anhui Provincial Key Laboratory of Network and Infomation Security, Wuhu 240002, China
[3] Zhongfu Information Co., Ltd., Jinan 250101, China; miaogx@zhongfu.net
* Correspondence: yuanlifeng@hdu.edu.cn

Abstract: In secret image sharing, the image is divided into several stego images, which are managed by corresponding participants. The secret image can be recovered only when the number of authorized participants is no less than the threshold. Thus, it is widely used to protect essential images, such as engineering drawings and product design drawings. In the traditional secret image sharing scheme, the threshold is fixed and unique. However, in practice, the security policy and the adversarial structure may change; therefore, the threshold must be adjusted dynamically. In this paper, we propose a novel secret image sharing scheme with a changeable threshold capability. Our scheme eliminates the limit of the changeable threshold and reduces the computation required. Also, our scheme is the first threshold changeable secret image sharing scheme that can recover an undistorted cover image. The theoretical analysis shows that our scheme is safe even if the threshold is changed. The experiments demonstrated that the stego image generated by our algorithm has better quality than other changeable-threshold, secret image sharing algorithms.

Keywords: symmetry; secret image sharing scheme; interpolation polynomial; threshold changeable

check for updates

Citation: Wu, G.; Wang, M.; Wang, Q.; Yao, Y.; Yuan, L.; Miao, G. A Novel Threshold Changeable Secret Image Sharing Scheme. *Symmetry* **2021**, *13*, 286. https://doi.org/10.3390/sym13020286

Academic Editor: Alice Miller
Received: 4 January 2021
Accepted: 4 February 2021
Published: 7 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of consumer and communication technologies, data, including images, increasingly are collected, transmitted, and disseminated only in electronic form. However, ensuring the security of images-in-transit and images-at-rest is challenging. In order to solve the security problem of private images, symmetric or public-key cryptography encryption images, which needs a key, are often used [1–3]. If the key is lost or the encrypted image is damaged, the secret information cannot be recovered. Secret sharing technology can provide private images with better security and fault tolerance. Using the secret image sharing technology, the private image is split into multiple shares and each share is distributed to authorized participants. Only a certain number of authorized participants can recover the original private image.

In 1979, Shamir [4] proposed a secret sharing scheme using the principle of polynomial interpolation. In the same year, Blakey [5] independently designed a secret sharing scheme based on space geometry. Since then, research on secret sharing has attracted the attention of researchers. In 1994, Naor and Shamir [6] proposed a secret image sharing scheme, generally, secure (t, n) image sharing schemes divide the secret image into n shares. No complicated calculations are required, and the secret image can be reconstructed by stacking t shares. However, the share, a random-noise-like image, can be noticed very easily by attackers. To reduce the probability of discovery, Lin and Tsai [7] proposed a new secret image sharing scheme, in which stenography is used to embed the secret share into the cover image. Then, their scheme uses watermarking to provide the capability to verify shares, thereby preventing share spoofing attacks. Secret sharing can effectively ensure the security of secret information, but the traditional secret sharing schemes (e.g., [8–12]) cannot adjust the threshold value. However, in reality, the threshold value must be adjusted

when the security policy or the adversary model changes. Here, we describe three example situations in which (1) the importance of the secret information must be changed, (2) some participants have left or joined the system, and (3) attackers have obtained one or more secret shares.

Inspired by the work of Yuan et al. [13], we propose a novel, adjustable, threshold image-sharing scheme. In our method, the threshold value is no longer fixed and can be adjusted dynamically and in a timely fashion according to the actual situation and security policy. By improving the polynomial generation method and the steganography of Guo et al.'s scheme [14], we achieve the following three features:

(1) There is no limit on the adjustable threshold values; therefore, the scheme can be used in more complex applications.

(2) By improving the polynomial generation algorithm, the secret can be recovered by calculating only one polynomial.

(3) We improved Guo et al.'s image hiding scheme [14] by using the location map to void the overflow pixel, which makes our scheme the first scheme to recover the cover image without distortion and to have the highest quality stego images compared to other threshold-changeable secret sharing schemes.

The remainder of the paper is organized as follows. In the next section, we introduce related work; in Section 3, we briefly revisit the work of Yuan et al. [13], before presenting our proposed scheme in Section 4; and in Section 5, we present our security and performance evaluations. Our conclusions are presented in the last section.

2. Related Work

In 1989, Laih et al. [15] proposed the first secret sharing scheme that had a changeable threshold. Desmedt and Jajodia's scheme [16] incorporated the use of the redistribution of shares, which removed the requirement for the dealer to be online. Before the secret can be recovered, participants must maintain a secure channel with each other. In 1999, Martin et al. [17] designed a general model of sharing secret information with adjustable thresholds. In 2005, Barwick et al. [18] proposed a new secret sharing scheme with changeable thresholds that used broadcast communication to minimize the costs. In recent years, researchers have used different technologies to implement secret sharing schemes with adjustable threshold values. For example, based on polynomial interpolation, in 2012, Zhang et al. [19] designed a secret sharing scheme with an adjustable threshold capability that also resists conspiratorial attacks launched by malicious participants using historical shares. In 2016, Yuan et al. [13] proposed a dealer-free adjustable threshold secret sharing scheme that was designed to defend against a historical share attack by introducing a two-variable function. In 2017, Pilaram et al. [20] proposed an adjustable-threshold multi-secret sharing scheme based on a lattice design. In 2018, Jia et al. [21] proposed a secret sharing scheme with an adjustable threshold. The core of the scheme was to construct a new prime-number matrix, and the generation and reconstruction of the shares were achieved using the Chinese remainder theorem.

With the wide application of images, researchers began to regard images as secret information and introduced secret sharing into the field of images. In 2007, Yang et al. [22] improved Lin and Tsai's [7] scheme by designing a scheme to achieve enhanced authentication using a Galois field. In a separate work, Lin and Chan [8] proposed an invertible secret image sharing scheme that achieved improved quality of the stego image, that provided larger embedding capacity, and that allowed for the reconstruction of an undistorted secret image. In 2012, Guo et al. [23] proposed the first hierarchical secret image sharing scheme. In their scheme, the stego images are partitioned into several levels, and each level has a corresponding threshold. The access structure is determined by a sequence of threshold requirements. Only when each threshold meets the current requirements and all of the thresholds are satisfied can the secret image be recovered without distortion. Hierarchical secret image sharing extends the boundary of secret image sharing, and such schemes are constantly being proposed [11,24,25]. To improve the quality of stego images, in 2013,

Ulutas et al. [26] proposed a novel secret image sharing scheme that uses the exploiting modification direction (EMD) technique and the modulus operator to hide shares. The approach ensures higher visual quality stego images for the cover image. In addition to hierarchical schemes, many researchers have proposed various multi-secret image sharing schemes based on different technologies. For example, the Chinese remainder theorem [10], Boolean [27–29], and cellular automaton [30] have been proposed. In 2017, Yuan et al. [31] first applied the adjustable threshold into secret image sharing, and they proposed the first adjustable-threshold secret image scheme. Their scheme can adjust the threshold values securely before recovering the secret image. However, their scheme has some limitations, e.g., the threshold value is conditional, the secret share requires large storage space, and a significant amount of calculation is required during recovery. By extending Thien and Lin's scheme [32], Liu et al. [33] proposed a threshold-changeable secret image sharing scheme in 2019. However, their scheme does not use image steganography and cannot resist collusion attacks.

3. Preliminaries

Here, we introduce the $(\{t_1, t_2, \ldots, t_N\}, n)$ adjustable-threshold secret sharing scheme of Yuan et al. [13], which is the two-variable one-way function used in our scheme.

3.1. Two-Variable One-Way Function

A two-variable one-way function $f(r, s)$ is a function of the variables r and s that can be mapped to a fixed range of values in a finite field. It has the following properties as described by Chien et al. [34]:

(1) Given r and s, it is easy to compute $f(r, s)$;
(2) Given s and $f(r, s)$, it is hard to compute r;
(3) Having no knowledge of s, it is hard to compute $f(r, s)$ for any r;
(4) Given s, it is hard to find two different values r_1 and r_2 such that $f(r_1, s) = f(r_2, s)$;
(5) Given r and $f(r, s)$, it is hard to compute s;
(6) Given pairs of r_i and $f(r_i, s)$, it is hard to compute $f(r_j, s)$ for $r_j \neq r_i$.

He and Dawson [35] introduced the concept of the two-variable one-way function through the existing one-way function and gave some theoretical proofs. In addition, they gave some examples of constructing two-variable one-way functions, such as using hash functions as follows: Let A be a secure signature scheme [36]. For a message m, the signature with secret key k is denoted by $A(k, m)$. Let h be a universal one-way hash function, the existence of which is based on any one-to-one, one-way function [37]. Let $f(x, y) = h(A(x, y))$. Then f is a two-variable one-way function that satisfies the properties of (1)–(6).

In our scheme, the two-variable one-way function is used mainly to defend against the historical share attack. At the same time, the real share, s_i, can be used in the next secret sharing process, which can improve efficiency.

3.2. Yuan et al.'s Secret Sharing Scheme with a Changeable Threshold

In 2016, Yuan et al. [13] proposed a novel $(\{t_1, t_2, \ldots, t_N\}, n)$-threshold secret sharing scheme. Their scheme does not have a limit on the value of the changeable threshold, and its computations are less complex. We briefly introduce the process of their scheme, which includes two procedures, i.e., the sharing procedure and the recovery procedure.

3.2.1. Sharing Procedure

To share secret s, the dealer constructs polynomial $h(x)$ as

$$h_N(x) = (s + a_1 x + a_2 x^2 + \ldots + a_{t_{N-1}} x^{t_{N-1}}) \bmod q \quad (1)$$

where q is a large prime number and $a_1, a_2, \ldots, a_{t_{N-1}} \in GF(q)$ are chosen randomly. Polynomial $h_j(x)$, which corresponds to the threshold $t_j (1 \leq j \leq N)$, can be generated as follows:

$$\begin{aligned} h_{(N-1)}(x) &= (s + a_1 x + \ldots + a_{t_{N-1}-1} x^{t_{N-1}-1}) \ mod \ q \\ h_{(N-2)}(x) &= (s + a_1 x + \ldots + a_{t_{N-2}-1} x^{t_{N-2}-1}) \ mod \ q \\ &\vdots \\ h_1(x) &= (s + a_1 x + \ldots + a_{t_1-1} x^{t_1-1}) \ mod \ q. \end{aligned} \tag{2}$$

Then, the dealer chooses different random integers, i.e., $r_1, r_2, \ldots, r_N$, corresponding to the thresholds $t_1, t_2, \ldots, t_N$ one to one. The shares can be calculated as follows:

$$y_i^j = h_j(f(r_j, s_i)) \ (1 \leq i \leq n, \ 1 \leq j \leq N), \tag{3}$$

where $f(r, s)$ is a two-variable one-way function and s_i is the real identification of the participant, P_i.

3.2.2. Recovering Procedure

Before recovering, if t_j participants or more than t_j participants want to recover the secret s, the combiner will broadcast the corresponding key, r_j. For a general description, we assume that participants $P_1, P_2, \ldots, P_{t_j}$ want to recover the secret. Then, the secret s can be recovered as follows:

$$s = h_j(0) = \sum_{i=1}^{t_j} y_i^j \prod_{k=1, k \neq i}^{t_j} \frac{-f(r_j, s_k)}{f(r_j, s_i) - f(r_j, s_k)} \ mod \ q. \tag{4}$$

4. Proposed Scheme

In this section, we introduce the process of our scheme, which is divided into two phases, i.e., the secret image sharing phase and the recovery phase.

In the secret sharing procedure, the dealer generates the secret shares from the secret image and embeds them in the cover image to form the stego images. Then, the dealer adjusts the threshold value according to the actual environment and security policy. If the number of authorized participants is more than or equal to the threshold value, it enters the recovery phase and reconstructs the secret image and cover image. The abstract flow of our scheme is shown in Figure 1, and the main notations of this paper are listed in Table 1.

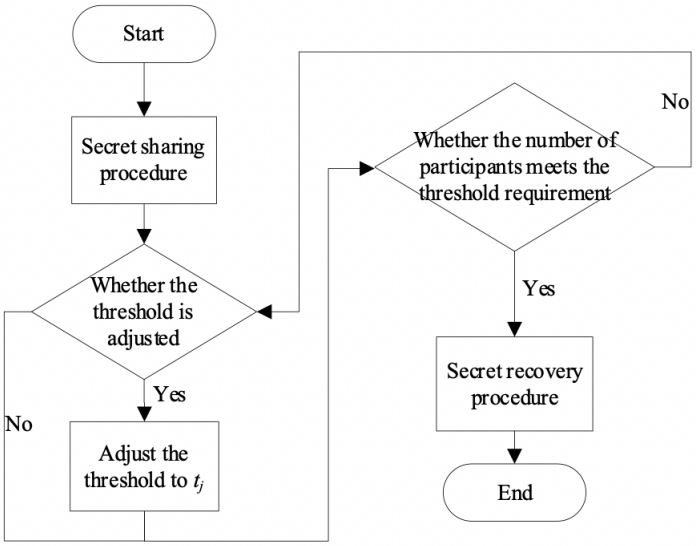


Figure 1. Abstract flow of our scheme.

Table 1. Summary of Notations.

Notation	Meaning
S	Gray-scale secret image S
C	Gray-scale cover image C
$sh \times sw$	sh and sw are the width and height of secret image S
$ch \times cw$	ch and cw is the width and height of cover image C
n	The number of participants
P_i	Participant i
S_i	The stego images holed by participant P_i
N	The number of changeable thresholds
t_j	The value of the jth changeable threshold
m	A prime number, and $m \in [0, 255]$
$h_j(x)$	The polynomial corresponding to threshold t_j
$f(r, s)$	A two-variable one-way function
s_i	The identification of participant P_i
r_j	The key corresponding to threshold t_j
$\left[x^k\right]$	Coefficient operator. If $h(x) = \sum_{i \geq 0} a_i x^i$, then $\left[x^k\right] h(x) = a_k$
$\lceil \cdot \rceil$	The ceiling function
$\lfloor \cdot \rfloor$	The flooring function
D	Converted data of secret image
R	Cover data of the cover image
M	Non-embedded location map m-ary data
L_i	Location map share data of participan P_i

4.1. Secret Sharing Procedure

In the secret sharing procedure, the dealer generates secret shares from secret image S and then embeds them in cover image C to form stego images $S_1, S_2, \ldots S_n$. The secret sharing procedure is divided into two parts, i.e., (1) the share generation phase and (2) the stego image generation phase. Figure 2 shows the flow diagram of the secret image sharing phase.

4.1.1. Share Generation Phase

In this phase, the secret image is processed to generate the shares of n participants. The work steps are as follows:

Step 1: According to the order from left to right and top to bottom, the dealer converts every pixel of secret image S into m-ary digits and forms the converted data D and $m \in [0, 255]$. Every pixel is converted to $\lceil log_m 255 \rceil$ digits.

Step 2: The dealer selects N changeable thresholds according to their needs and in ascending order $(t_{i-1} < t_i, 2 \leq i \leq N)$.

Step 3: The dealer selects t_1 digits $d_0, d_1, \ldots, d_{t_1-1}$ from converted data D and constructs original polynomial $h_N(x)$ as follows:

$$h_N(x) = a_0 + a_1 x + \ldots + a_{t_N-1} x^{t_N-1} \; mod \; m, \tag{5}$$

where $a_i = d_i (0 \leq i \leq t_1 - 1)$ and $t_N - t_1$ different integers $a_{t_1}, a_{t_1+1}, \ldots, a_{t_N-1}$ are chosen randomly in $GF(m)$.

Step 4: According to Algorithm 1, the polynomials $h_1(x), h_2(x), \ldots, h_{N-1}(x)$ corresponding to the thresholds $t_1, t_2, \ldots, t_{N-1}$ can be generated as follows:

$$\begin{aligned} h_{(N-1)}(x) &= (a_0 + a_1 x + \ldots + a_{t_{N-1}-1} x^{t_{N-1}-1}) \; mod \; m \\ h_{(N-2)}(x) &= (a_0 + a_1 x + \ldots + a_{t_{N-2}-1} x^{t_{N-2}-1}) \; mod \; m \\ &\vdots \\ h_1(x) &= (a_0 + a_1 x + ... + a_{t_1-1} x^{t_1-1}) \; mod \; m. \end{aligned} \tag{6}$$

Step 5: The dealer selects n distinct and nonzero random integers $s_1, s_2, \ldots, s_n$ to identify participants $P_1, P_2, \ldots, P_n$, and the dealer randomly selects N different integers $r_1, r_2, \ldots, r_N$ as the key. Then, shares $y_i^1, y_i^2, \ldots, y_i^N (1 \leq i \leq n)$ can be calculated as follows:

$$y_i^j = h_j(f(r_j, s_i))(1 \leq j \leq N), \tag{7}$$

where y_i^j is participant P_i's share corresponding to the jth threshold and $f(r, s)$ is a function with two-variable one-way.

Step 6: Repeat steps 3–5 until the converted data are completely processed.

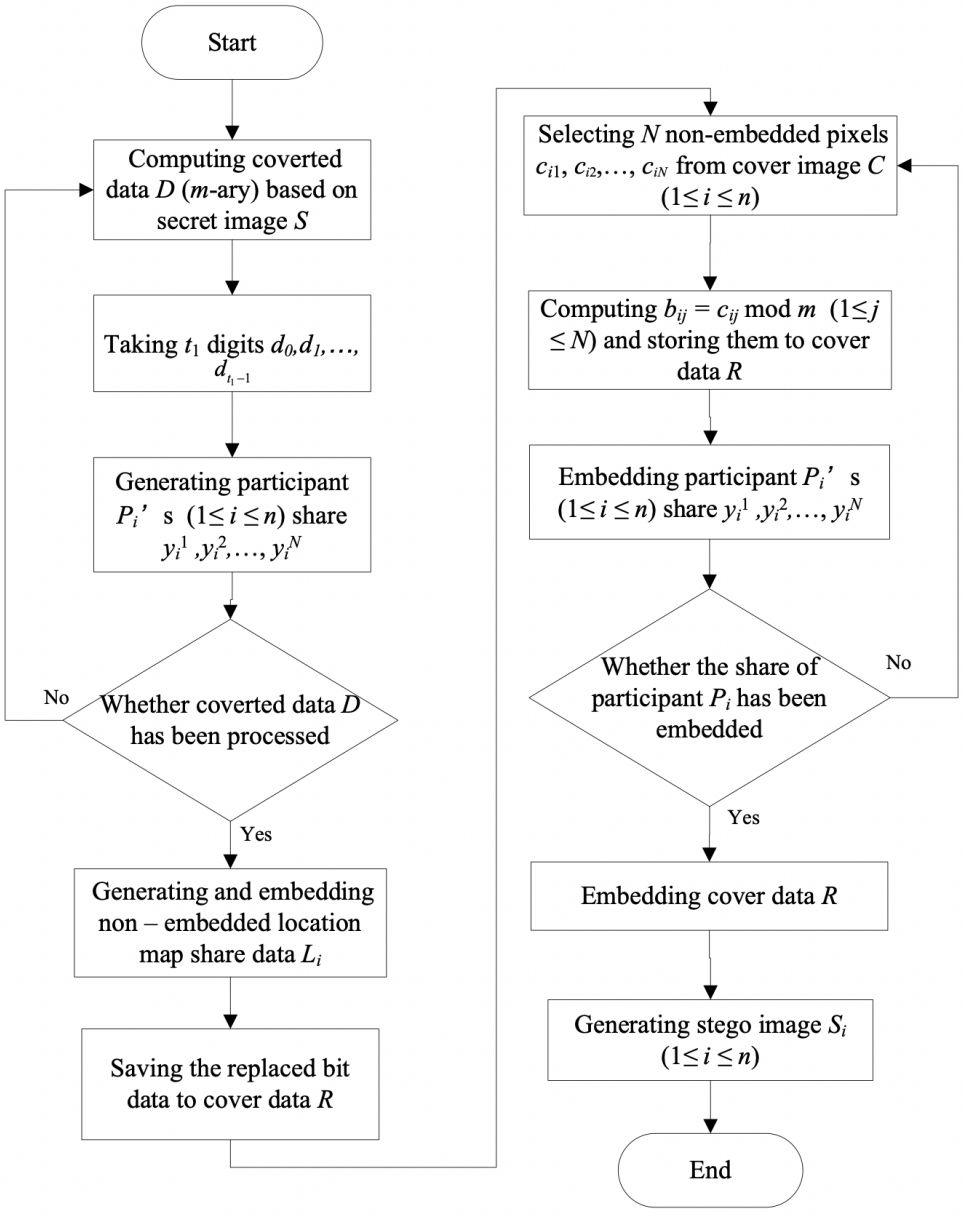


Figure 2. Flow diagram of the secret image sharing phase.

Algorithm 1 Polynomial generator.

Input: $h_N(x), j, t_j, t_N$

Output: $h_j(x)$

$h_j(x) = h_N(x)$;

$d = t_N - t_j$;

while $d > 0$ **do**

$c = [x^{t_j+d-1}]h(x)$

$h_j(x) = h_j(x) - cx^{t_j+d-1}$

end while

4.1.2. Stego Images Generation Phase

In this phase, the dealer embeds every participant's share into the cover image and generates a corresponding stego image. The steps are as follows.

Step 1: Non-location map generation and hiding

The embedded pixel value may overflow (The reason is explained in Section 5); therefore, we need to generate the non-embedded location map. According to the principle of symmetry, a non-embedded location map is generated for the cover image as follows. First, we set a map of the same size as the cover image, and the default value for each position in the map is 0. Then, if the pixel c, which comes from cover image C, is not within the range of $[\lceil (m-1)/2 \rceil, 255 - \lfloor (m-1)/2 \rfloor]$, the corresponding position in map is modified to 1. As shown in Figure 3, we can obtain a corresponding non-embedded location map based on the 3×3 block from the gray-scale image "Crowd", where $m = 7$ and the range of embeddable pixels is [3, 252].

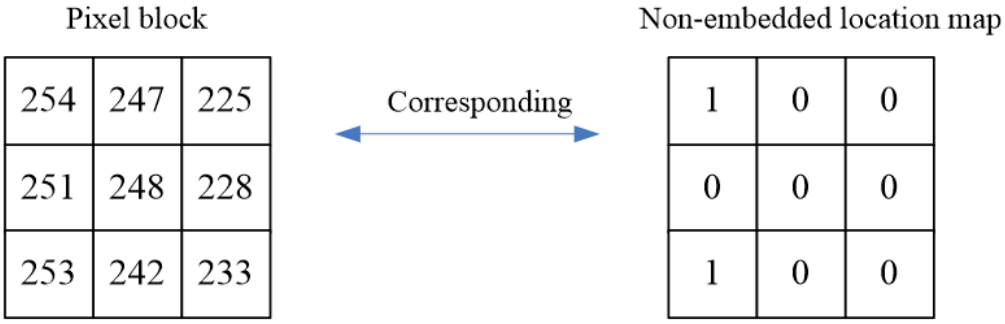


Figure 3. Example of generating a non-embedded location map when m = 7.

In order to reduce the amount of embedded data, we used the secret image sharing method to generate shares of the non-embedded location map data as follows:

① The non-embedded location map is converted to m-ary data (called location map data M).

② The dealer selects t_1 digits $m_0, m_1, \ldots, m_{t_1-1}$ from data M and constructs the following polynomial:

$$g(x) = m_0 + m_1 x + \ldots + m_{t_1-1} x^{t_1-1} \ mod \ m. \tag{8}$$

Then, the share $g(s_i)(1 \leq i \leq n)$ of participant P_i is calculated, where s_i is participant P_i's identification. Share $g(s_i)$ is converted to binary data and saved to location map share data L_i.

③ Repeat step ② until data M are processed.

④ We used the LSB(Least Significant Bit) algorithm to embed L_i into the cover image. Assume that the w-bit data in the pixel are replaced with the location map share data. The dealer selects unembedded pixel c from cover image C in the order from left to right and

top to bottom and saves $c_{w-1}, c_{w-2}, \ldots, c_0$ to cover data R, where pixel c is represented in 8-bit binary as $c_7, c_6, \ldots, c_0$. Then, the dealer selects w digits $l_0, l_1, \ldots, l_{w-1}$ from the location map share data, L_i, and replaces $c_{w-1}, c_{w-2}, \ldots, c_0$ with $c_i = l_{w-1-i} (0 \leq i \leq w-1)$.

⑤ Repeat step ④ until the location map share data, L_i, are embedded.

Step 2: Hiding Shares

According to the non-embedded location map, the dealer selects N embeddable pixels $c_1, c_2, \ldots, c_N$ from cover image C. Then, the dealer computes $b_1, b_2, \ldots, b_N$ and saves them to cover data R, where $b_j = c_j \mod m$ $(1 \leq j \leq N)$. The pixel values, i.e., $sp_i^1, sp_i^2, \ldots, sp_i^N$ $(1 \leq i \leq n)$, used to replace $c_1, c_2, \ldots, c_N$ can be calculated as

$$sp_i^j = \begin{cases} c_j' - m & if(-m < \sigma_i^j < -\left\lfloor \frac{m-1}{2} \right\rfloor) \\ c_j' & if(-\left\lfloor \frac{m-1}{2} \right\rfloor \leq \sigma_i^j \leq \left\lceil \frac{m-1}{2} \right\rceil), \\ c_j' + m & if(\left\lceil \frac{m-1}{2} \right\rceil < \sigma_i^j < m) \end{cases} \tag{9}$$

where $c_j' = c_j - b_j + y_i^j$, and $\sigma_i^j = b_j - y_i^j$, $y_i^1, y_i^2, \ldots, y_i^N$ are the shares of participant P_i $(1 \leq i \leq n)$.

Then, the dealer repeats the above processes until all of the shares of data are embedded.

Step 3: Cover data hiding

To recover the cover image without distortion, cover data R must be embedded into the cover image. The dealer selects an unembedded pixel c from cover image C and calculates $b = c \ mod \ m$. Then, the dealer selects $t_1 - 1$ digits $r_0, r_1, \ldots, r_{t_1-1} \in GF(m)$ from cover data R and constructs polynomial $g(x)$ as

$$g(x) = r_0 + r_1 x + ... + r_{t_1-2} x^{t_1-2} + b x^{t_1-1} \ mod \ m. \tag{10}$$

Then, the dealer computes every participant's share $g_i = g(s_i)$, where s_i is the identification of participant $P_i (1 \leq i \leq n)$ and $c' = c - b + g_i$. Stego image pixel sp_i is calculated as

$$sp_i = \begin{cases} c' - m & if(-m < b - g_i < -\left\lfloor \frac{m-1}{2} \right\rfloor) \\ c' & if(-\left\lfloor \frac{m-1}{2} \right\rfloor \leq b - g_i \leq \left\lceil \frac{m-1}{2} \right\rceil). \\ c' + m & if(\left\lceil \frac{m-1}{2} \right\rceil < b - g_i < m) \end{cases} \tag{11}$$

The dealer repeats the above processes until cover data R are embedded into cover image C, and then, the dealer generates all stego image $S_1, S_2, \ldots, S_n$. The composition of the stego image is shown in Figure 4.

Last, the dealer sends stego image S_i and identification s_i to participant P_i through the secure channel and destroys secret image S and all of the stego images, i.e., $S_1, S_2, \ldots, S_n$, to avoid a situation in which the attacker can attain the secrete image S by attacking the dealer.

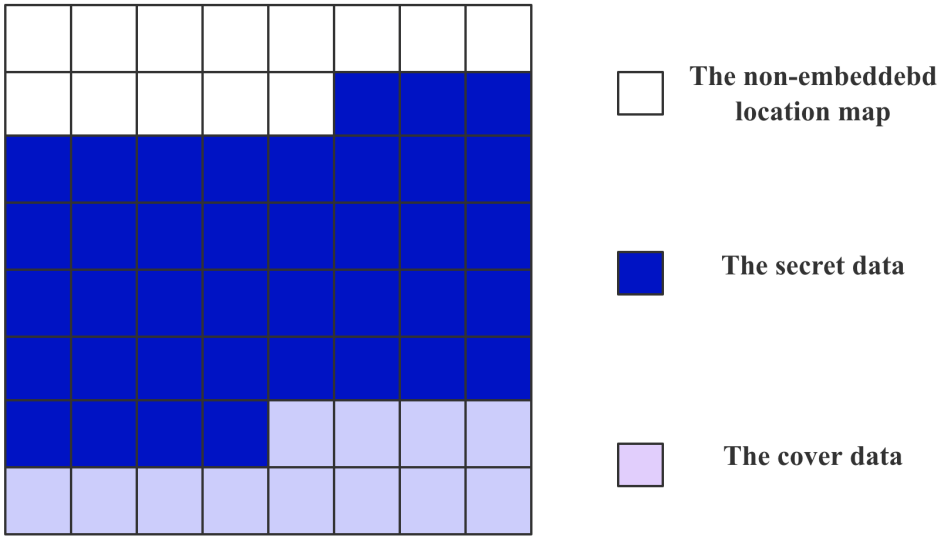


Figure 4. Composition of the stego image.

4.2. Recovery Procedure

When the security strategy or the adversary model changes, the threshold must be adjusted to maintain the original security level. Assume that the threshold value is changed to t_j $(1 \leq j \leq N)$. Then, if the number of participants who agree to join in the recovery phase is greater than or equal to t_j, the dealer broadcasts key r_j and the secret image S can be recovered.

Without loss of generality, we assume that participants $P_1, P_2, \ldots, P_{t_j}$ agree to recover secret image S. Then, the authorized participants recover secret image S and cover image C through three steps, i.e., (1) extraction of the non-embedded location map, (2) reconstruction of the secret image, and (3) recovery of the cover image. Figure 5 shows the flow diagram of the recovery phase.

Step 1: Extraction of the non-embedded location map

① The authorized participants select $t_1 (t_1 \leq t_j)$ stego pixels from corresponding stego images $S_1, S_2, \ldots, S_{t_1}$(note: since $t_1 = \min\{t_1, t_2, \ldots, t_N\}$, at least t_1 participants participate in the recovery phase regardless of the current threshold).

② For each stego image S_i $(1 \leq i \leq t_1)$, the authorized participants select pixel sp_i in the order from left to right and top to bottom and represent pixel sp_i as $c_i^7, c_i^6, \ldots, c_i^0$. Then, they extract the embedding data $c_i^{w-1}, c_i^{w-2}, \ldots, c_i^0$.

③ Repeat step ② to obtain the location map share data L_i.

④ The share $g(s_i)$ of stego image S_i is extracted from L_i. Then, the polynomial $g(x)$ can be reconstructed as [38]:

$$g(x) = \sum_{i=1}^{t_1} g(s_i) \prod_{j=1, j \neq i}^{t_1} \frac{x - s_i}{s_i - s_j} \ mod \ m. \tag{12}$$

From the coefficient of polynomial $g(x)$, we can obtain the recovery digits, i.e., $m_0, m_1, \ldots, m_{t_1-1}$.

⑤ Repeat step ④ until the non-embedded data of the location map are recovered.

Step 2: Reconstruction of the secret image

According to the extracted, non-embedded location map, the corresponding stego pixels, i.e., $sp_1^j, sp_2^j, \ldots, sp_{t_j}^j$, can be obtained from stego images $S_1, S_2, \ldots, S_{t_j}$. By using

keys r_j and t_j and the participants' identifications, i.e., $s_1, s_2, \ldots, s_{t_j}$, the polynomial $h_j(x)$ can be reconstructed as [38]

$$h_j(x) = \sum_{i=1}^{t_j} y_i^j \prod_{k=1,k\neq i}^{t_j} \frac{x - f(r_j, s_k)}{f(r_j, s_i) - f(r_j, s_k)} \ mod\ m, \tag{13}$$

where $y_i^j = sp_i^j \bmod m$.

By repeating the processes above, all converted data, D, can be extracted, and the secret image, S, can be recovered by converting data, D.

Step 3: Recovery of the cover image

The authorized participants select $t_1 (t_1 \leq t_j)$ stego pixels from the corresponding stego images. We assumed that the authorized participants selected stego pixels $sp_1, sp_2, \ldots, sp_{t_1}$ from the stego images $S_1, S_2, \ldots, S_{t_1}$. Then, polynomial $g(x)$ can be reconstructed as

$$g(x) = \sum_{i=1}^{t_1} g(s_i) \prod_{j=1,j\neq i}^{t_1} \frac{x - s_i}{s_i - s_j} \ mod\ m, \tag{14}$$

where $g(s_i) = sp_i \mod m$. From the coefficient of polynomial $g(x)$, we can obtain cover digits $r_0, r_1, \ldots, r_{t_1-2}$ and b. Digits $r_0, r_1, \ldots, r_{t_1-2}$ can be used to recover the cover image pixels of the location map area and the share area, and digit b can be used to recover the cover image pixel of the cover data area. Then, the participants repeat these processes to obtain cover data R. Because the LSB algorithm normally is used, we only represent how to recover cover pixels used to embed the share and the cover data.

Assume that we use stego images $S_1, S_2, \ldots, S_{t_j}$ to recover the area of cover image C used to embed the share. Then, we select N stego image pixels $sp_i^1, sp_i^2, \ldots, sp_i^N$ from the corresponding area of stego image $S_i (1 \leq i \leq t_j)$ and select N digits $r_1, r_2, \ldots, r_N$ from cover data R. As we know, $b_j = c_j \bmod m$, where c_j is the cover image pixel value corresponding to stego image pixel sp_i^j. (See the details in Section 4.1) Stego image pixel $sp_i^j (1 \leq j \leq N)$ can be computed as

$$sp_i^j = \begin{cases} c_j' - m & if(-m < \sigma_i^j < -\left\lfloor \frac{m-1}{2} \right\rfloor) \\ c_j' & if(-\left\lfloor \frac{m-1}{2} \right\rfloor \leq \sigma_i^j \leq \left\lceil \frac{m-1}{2} \right\rceil), \\ c_j' + m & if(\left\lceil \frac{m-1}{2} \right\rceil < \sigma_i^j < m) \end{cases} \tag{15}$$

where $c_j' = c_j - b_j + y_i^j$ and $\sigma_i^j = b_j - y_i^j$, where $y_i^j = sp_i^j \bmod m$. Thus, the cover image pixel c_j can be recovered as

$$c_j = \begin{cases} sp_i^j + \sigma_i^j + m & if(-m < \sigma_i^j < -\left\lfloor \frac{m-1}{2} \right\rfloor) \\ sp_i^j + \sigma_i^j & if(-\left\lfloor \frac{m-1}{2} \right\rfloor \leq \sigma_i^j \leq \left\lceil \frac{m-1}{2} \right\rceil), \\ sp_i^j + \sigma_i^j - m & if(\left\lceil \frac{m-1}{2} \right\rceil < \sigma_i^j < m). \end{cases} \tag{16}$$

After recovering the cover pixel used to embed the share, we can use the same method to recover the cover pixel used to embed the cover data. Then, we can recover the cover image without distortion.

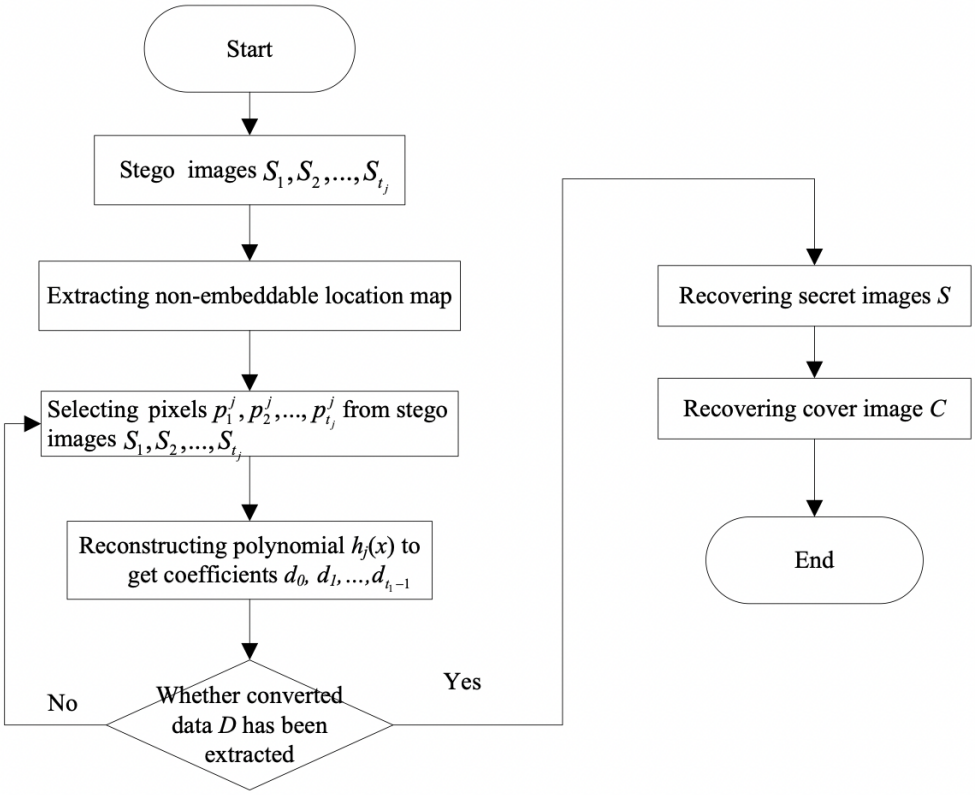


Figure 5. Flow diagram of the recovery phase.

5. Experiment and Analysis

In this section, we describe the experiments that were conducted on the scheme. Then, the performance of the experiments are analyzed in detail, and finally, the security of the scheme is discussed.

In the experiments, we use the peak signal-to-noise ratio (PSNR) as a measure of performance [39]:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right) \mathrm{dB}. \tag{17}$$

The mean square error (MSE) is defined in an $H \times W$-sized image [39]:

$$\mathrm{MSE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(p_{ij} - p'_{ij} \right)^2, \tag{18}$$

where p_{ij} is the original pixel value before embedding and p'_{ij} is the pixel value of the stego image.

5.1. Simulation Results

In our experiment, the parameters were set as follows: $N = 3$, $n = 6$, $m = 7$, $t_1 = 3$, $t_2 = 4$, and $t_3 = 5$, which is a $(\{3, 4, 5\}, 6)$ threshold-changeable secret image sharing scheme. We chose a gray-scale baboon of 128×128 pixels as the secret image, as shown in Figure 6.

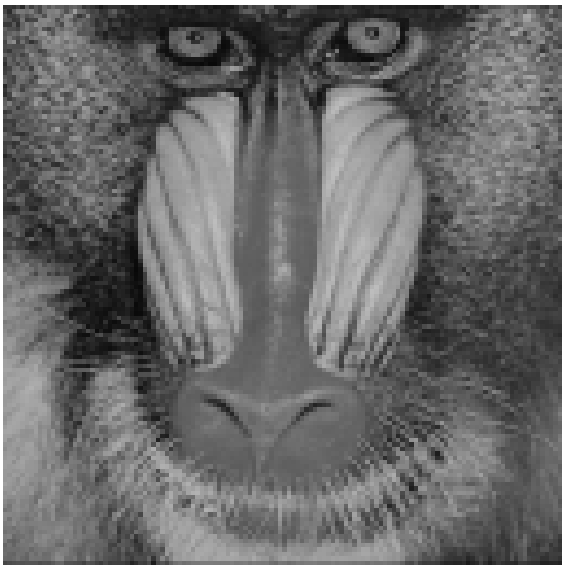


Figure 6. Secret image of a baboon.

Initially, we used the classic standard test image, Lena, at 512×512 pixels, as the cover image. In our $(\{3,4,5\},6)$ scheme, six stego images were obtained, and then, $t_2 = 4$ was used as the changed threshold and the key, r_2, was broadcast. For simplicity, we choose to use images (a), (b), (c), and (d) to recover the secret image, and Figure 7 shows the experimental results, i.e., the average PSNR of the stego image was 47.15 dB. The results showed that this scheme has a good quality image, i.e., the difference between the stego image and the cover image could not be detected by the human eye, and the secret image can be recovered without any distortion.

(a) Stego image 1, PSNR = 47.14 dB

(b) Stego image 2, PSNR = 47.13 dB

(c) Stego image 3, PSNR = 47.16 dB

(d) Stego image 4, PSNR = 47.17 dB

(e) Stego image 5, PSNR = 47.13 dB

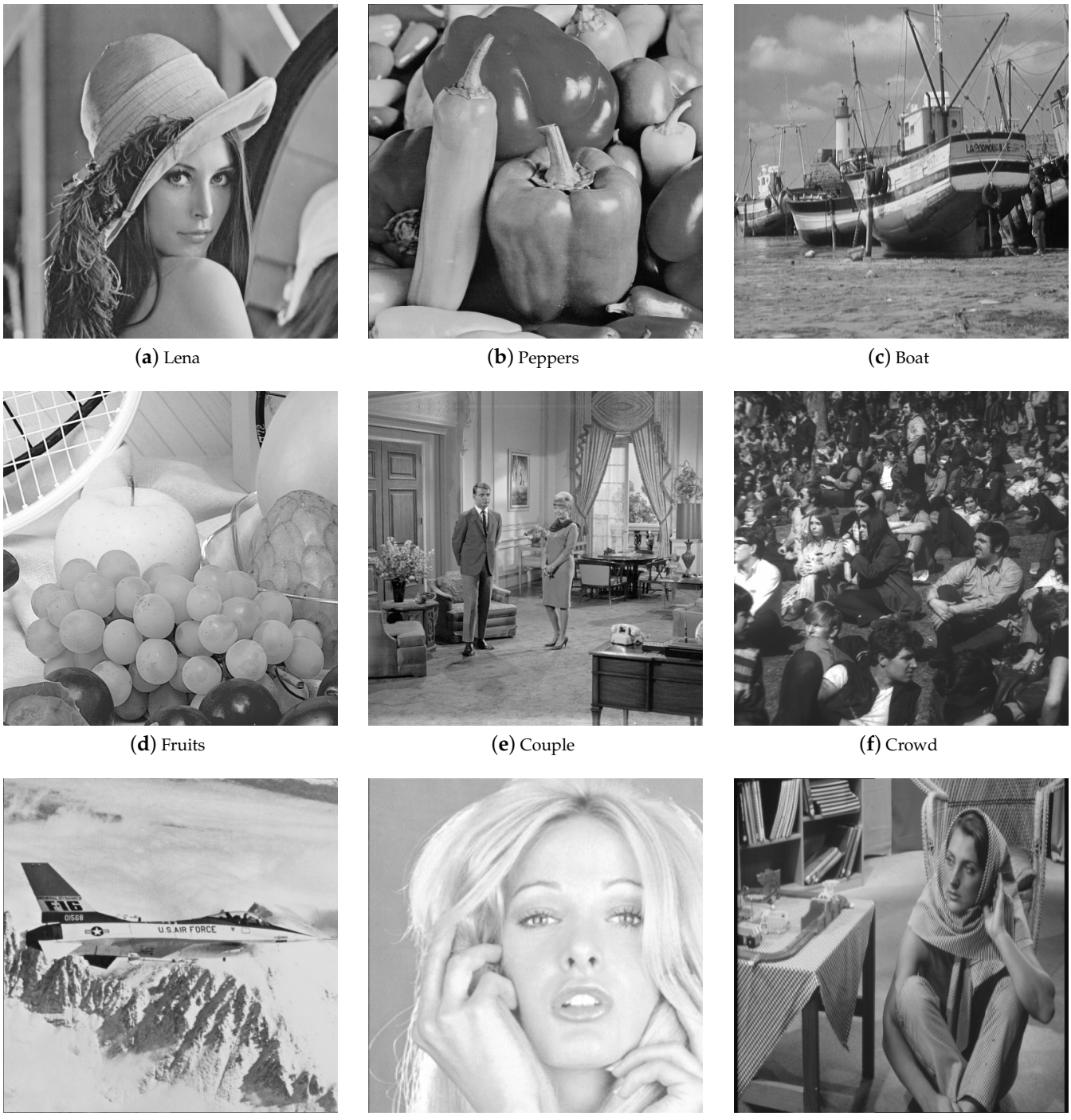
(f) Stego image 6, PSNR = 47.14 dB

(g) Cover image

(h) Recoverd secret

Figure 7. Stego images, recovered secret image baboon, and cover image Lena.

In addition, the influence of the cover image on the quality of the stego image was considered. We used 9 different 512 × 512-pixel gray-scale images as the cover image. This group of images is shown in Figure 8, and the experimental results are shown in Table 2. It was apparent that the difference between the results of different cover images was small; therefore it was concluded that the cover image has little effect on the quality of the stego images.



(**a**) Lena (**b**) Peppers (**c**) Boat

(**d**) Fruits (**e**) Couple (**f**) Crowd

(**g**) Airplane (**h**) Tiffany (**i**) Barbara

Figure 8. Nine cover images.

Table 2. Peak signal-to-noise ratio (PSNR) values (dB) of the stego images tested in nine cover images.

Cover Images	Stego 1	Stego 2	Stego 3	Stego 4	Stego 5	Stego 6	Average
Lena	47.14	47.13	47.16	47.17	47.13	47.14	47.15
Peppers	47.20	47.19	47.22	47.20	47.23	47.22	47.21
Boat	47.22	47.13	47.22	47.24	47.15	47.20	47.19
Fruits	47.03	47.12	47.00	47.14	47.03	47.10	47.07
Couple	47.15	47.22	47.22	47.13	47.15	47.22	47.18
Crowd	47.17	47.16	47.15	47.17	47.13	47.15	47.15
Airplane	47.11	47.12	47.13	47.13	47.12	47.11	47.12
Tiffany	47.03	47.14	47.11	47.14	47.06	47.03	47.08
Barbara	47.16	47.19	47.19	47.18	47.17	47.20	47.18

5.2. Performance Analysis

In this section, we focus on analyzing the parameter m and thresholds $t_1, t_2, \ldots, t_N$ because these two parameters have important influences on the secret capacity and the quality of secret images in our scheme.

For the analysis of m, our experimental parameters are as follows: $N = 3$, $n = 6$, $t_1 = 3$, $t_2 = 4$, and $t_3 = 5$. The baboon is still the secret image, and Lena is used as the cover image; Table 3 shows the results of the experiment, i.e., the relationship between the different values of m and the secret capacity. Based on the experimental data, we concluded the following: (1) as m increases, the value of PSNR decreases; (2) the secret embedding capacity increases as the factor m increases. For the first point, we can obtain it from the range of pixels. As the range of pixels changes and becomes larger, the difference between the pixels of the stego image and the pixels of the cover image increases; therefore, the PSNR value increases. Also, as the factor m increases, there is a lower share of data after conversion. That is to say, there are fewer areas on the cover image where changes occurred; therefore, the embeddable capacity increases. In our scheme, there were t_1 converted digits in the polynomial that was constructed, and it was embedded in N cover pixels (i.e., $N \times \lceil log_m 255 \rceil / t_1$ pixel); similarly, $t_1 - 1$ cover data occupy one cover pixel. For a cover image with $H_c \times W_c$ pixels and considering the embedded pixels of the location map (the size of which is $\left\lceil \left\lceil \frac{log_m H_c \times W_c}{t_1} \right\rceil \times \frac{log_2 m}{w} \right\rceil$), the embedding capacity is $\frac{t_1(t_1-1)}{\lceil N(t_1-1)+t_1 \rceil \times \lceil log_m 255 \rceil} \times (H_c \times W_c - \left\lceil \left\lceil \frac{log_m H_c \times W_c}{t_1} \right\rceil \times \frac{\lceil log_2 m \rceil}{w} \right\rceil)$.

Table 3. Relationship of the capacity distortion for different m values.

m	Pixel Change Range	PSNR
7	$[-3, 3]$	47.15
11	$[-5, 5]$	43.32
13	$[-6, 6]$	41.81
17	$[-8, 8]$	39.33
19	$[-9, 9]$	38.34
23	$[-11, 11]$	36.55

The value of the thresholds is another important factor in the scheme. We chose different thresholds in the experiment, and Table 4 shows the results of the experiment. In the experiment, there were five different sets of potential thresholds, i.e., $\{2,3,4\}$, $\{2,4,7\}$, $\{3,4,5\}$, $\{3,5,7\}$, $\{6,7,8\}$, and $m = 7$; other parameters were set as above.

In our scheme, t_1 data were embedded for each secret sharing; therefore, the greater the threshold t_1, the greater the embedding capacity. The data in Table 4 are consistent with our theoretical analysis. Analyzing the thresholds $\{2,3,4\}$, $\{2,4,7\}$, $\{3,4,5\}$, and $\{3,5,7\}$ with fixed m and N, it was apparent that the quality of the image had little relationship with t_N. It is worth noting that the value of the minimum threshold and the value of the

potential threshold are important factors that determine the quality of the stego images. This is because the minimum threshold and the number of thresholds affect the amount of embedded data. Then, by analyzing the data that correspond to the numbers 1, 3, and 5 in Table 4, it was found that the PSNR increases gradually as the factor, t_1, increases. The increase in the embedding capacity means fewer shares of fixed secret data, i.e., the number of pixels in the original cover image is modified less; therefore, the quality of the stego image is better.

Table 4. Performance comparisons of different potential thresholds.

Number	Thresholds	PSNR (dB)
1	{2, 3, 4}	44.25
2	{2, 4, 7}	44.24
3	{3, 4, 5}	47.15
4	{3, 5, 7}	47.15
5	{6, 7, 8}	51.18

Table 5 shows the comparison of our scheme with other secret image sharing schemes in recent years. We still used the baboon with 256×256 pixels as the secret image and Lena with 1024×1024 pixels as the cover image and set the thresholds as $\{2, 3, 4\}$, $m = 5$. The results shown in Table 5 indicate that the PSNR value of our scheme was slightly lower than the PSNR values of the single-threshold, secret image sharing schemes. This occurred because our scheme embedded redundant shares and non-embedded location map information. However, in all secret image sharing schemes with changeable thresholds, our scheme had the best quality images. Although the embedding method was optimized, a location map also was embedded in order to recover the cover image; therefore, compared with Yuan et al.'s scheme [31], the improvement in the quality of the stego image was not obvious. In addition, since Liu et al.'s scheme [33] does not use steganography to hide the shared data, the shadow image is generated directly in their scheme; therefore, there is no meaningful reference to PSNR.

We also compared and analyzed the actual running performance. Considering that different types of secret image sharing schemes were not suitable for comparison, we experimented on the same type schemes, such as Yuan et al.'s scheme and Liu et al.'s scheme. The secret image was the baboon with 128×128 pixels, and the cover image was Lena with 512×512 pixels. The other parameters were $N = 3$, $t_1 = 3$, $t_2 = 4$, $t_3 = 5$, $m = 7$, and $n = 6$. The adjusted threshold value $t_2 = 4$ was chosen for the actual performance test. The execution time of Liu et al.'s scheme is 32 s, that of Yuan et al.'s scheme is 151 s, and that of our scheme is 147 s. Liu et al.'s scheme does not have the steganography operation; therefore, its execution time is shorter than our scheme. Yuan et al.'s scheme needs to recover multiple polynomials layer by layer to recover the secret image, but our scheme only needs to compute a polynomial. Meanwhile, our scheme needs to generate and embed a location map, which Yuan et al.'s scheme does not need. Thus, our scheme's execution time is shorter but close to Yuan et al.'s. Compared with other methods, our work has the following advantages and contributions:

(1) No limit on thresholds. In Yuan et al.'s scheme [31], the threshold can be changed only once. However, in our scheme, N potential thresholds do not need to satisfy $t_{i+1} - t_i \leq t_1 (i = 1, 2, \ldots, N-1)$. Meanwhile, the threshold value of our scheme can be changed more than once.

(2) Less calculation. In Yuan et al.'s scheme [26], if the threshold is adjusted to $t_j (1 \leq j \leq N)$, they must use polynomial interpolation to determine the polynomial $h_j(x)$ and then to determine the polynomial $h_{j+1}(x)$,$h_{j+2}(x)$, …, $h_N(x)$ to obtain the secret from the polynomial $h_N(x)$. When recovering, our scheme does not have the process of iterating the polynomials, and only one polynomial has to be recovered.

(3) Recoverable cover image. In the same type of scheme, Yuan et al.'s scheme [31] cannot recover the cover image and Liu et al.'s scheme [33] does not use steganography;

however, our proposed method can reconstruct the cover image and completely recover the secret image without distortion.

5.3. Security Analysis

In this part, we prove the applicable pixel range of the proposed method, and the security of the scheme is analyzed theoretically.

Theorem 1. *In our scheme, the difference Δ_i^j between original pixel c_j and the corresponding stego pixel sp_i^j must satisfy $-\lceil (m-1)/2 \rceil \leq \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$, where m is a prime number and $m \in [0, 255]$.*

Proof. In our scheme, stego pixel sp_i^j is calculated as

$$sp_i^j = \begin{cases} c_j' - m & if(-m < \sigma_i^j < -\left\lfloor \frac{m-1}{2} \right\rfloor) \\ c_j' & if(-\left\lfloor \frac{m-1}{2} \right\rfloor \leq \sigma_i^j \leq \left\lceil \frac{m-1}{2} \right\rceil), \\ c_j' + m & if(\left\lceil \frac{m-1}{2} \right\rceil < \sigma_i^j < m) \end{cases} \tag{19}$$

where $c_j' = c_j - b_j + y_i^j$, $\sigma_i^j = b_j - y_i^j$, and $y_i^j = sp_i^j \bmod m$. Then, there are three cases to consider:

(1) $-m < b_j - y_i^j < -\lfloor (m-1)/2 \rfloor$

If there exists $-m < b_j - y_i^j < -\lfloor (m-1)/2 \rfloor$, then, $\Delta_i^j = sp_i^j - c_j = -b_j + y_i^j - m$. Since $-m < b_j - y_i^j < -\lfloor (m-1)/2 \rfloor$, it is easy to prove that $\lfloor (m-1)/2 \rfloor - m < \Delta_i^j < 0$. The equation can be expressed as $-\lceil (m-1)/2 \rceil \leq \Delta_i^j < 0$ because Δ_i^j is an integer.

(2) $-\lfloor (m-1)/2 \rfloor \leq b_j - y_i^j \leq \lceil (m-1)/2 \rceil$

The difference Δ_i^j can be expressed as $\Delta_i^j = sp_i^j - c_j = -b_j + y_i^j$ when$-\lfloor (m-1)/2 \rfloor \leq b_j - y_i^j \leq \lceil (m-1)/2 \rceil$. Thus, taking the opposite of $-\lfloor (m-1)/2 \rfloor \leq b_j - y_i^j \leq \lceil (m-1)/2 \rceil$, we can conclude that $-\lceil (m-1)/2 \rceil \leq \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$.

(3) $\lceil (m-1)/2 \rceil < b_j - y_i^j < m$

The difference Δ_i^j can be expressed as $\Delta_i^j = sp_i^j - c_j = -(b_j - y_i^j) + m$ when$\lceil (m-1)/2 \rceil < b_j - y_i^j < m$. Then, we can obtain $0 < \Delta_i^j < m - \lceil (m-1)/2 \rceil$, which can be transferred to $0 < \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$ because Δ_i^j is an integer. □

In summary, we can prove that $-\lceil (m-1)/2 \rceil \leq \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$.

Theorem 2. *Let the original pixel, c_j, satisfy $\lceil (m-1)/2 \rceil \leq c_j \leq 255 - \lfloor (m-1)/2 \rfloor$, where $m \in [0, 255]$ is a prime number. Then, we can ensure that the generated stego pixel, sp_i^j, satisfies $sp_i^j \in [0, 255]$.*

Proof. In view of Theorem 1, there exists $-\lceil (m-1)/2 \rceil \leq \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$, where $\Delta_i^j = sp_i^j - c_j$. Since $sp_i^j \in [0, 255]$, there are two cases to consider:

(1) $sp_i^j \leq 255$

In this case, sp_i^j can be expressed as $sp_i^j = c_j + \Delta_i^j$. Thus, the original pixel c_j must satisfy $c_j \leq 255 - \lfloor (m-1)/2 \rfloor$ because $-\lceil (m-1)/2 \rceil \leq \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$ and $c_j + \Delta_i^j \leq 255$.

(2) $sp_i^j \geq 0$

Since $-\lceil (m-1)/2 \rceil \leq \Delta_i^j \leq \lfloor (m-1)/2 \rfloor$ and$c_j + \Delta_i^j \geq 0$, we can prove that the original pixel, c_j, must satisfy $c_j \geq \lceil (m-1)/2 \rceil$.

To summarize, the original pixel, c_j, must satisfy $\lceil (m-1)/2 \rceil \leq c_j \leq 255 - \lfloor (m-1)/2 \rfloor$, which can ensure that stego pixel sp_i^j calculated through formula (19) does not exceed the range [0, 255]. □

According to Theorem 2, we embed the data into the original pixels that satisfy $\lceil (m-1)/2 \rceil \leq c_j \leq 255 - \lfloor (m-1)/2 \rfloor$ and use the non-embedded location map to record the original pixels that do not meet this condition.

Theorem 3. *Before broadcasting the key, no participant can calculate her/his share point pairs, which can be used to recover the polynomial by Lagrange interpolation.*

Proof. Assuming that the current threshold is $t_j \in \{t_1, t_2, \ldots, t_N\}$, the corresponding key is r_j $(1 \leq j \leq N)$, and participant $P_i (1 \leq i \leq n)$ wants to recover her/his first share point pair $\left(x_i^j, y_i^j\right)$, where $x_i^j = f(r_j, s_i)$, and s_i is P_i's identification. According to feature (2) of the two-variable one-way function, i.e., $f(r, s)$ presented in Section 3.1, participant P_i cannot calculate $(f(r_j, s_i), y_i^j)$ without key r_j. Similarly, participant P_i cannot recover the rest of her/his share point pairs and the other participants also cannot recover their share point pairs. Thus, no participant can obtain her/his share point pairs before broadcasting the key. □

Theorem 4. *If the number of authorized participants is less than the current threshold, the secret image cannot be recovered.*

Proof. According to the people who take part in the recovery phase, there are two cases to consider when recovering:

(1) Only authorized participants take part in the recovery phase.

Assuming that the threshold is $t_j \in \{t_1, t_2, \ldots, t_N\}$, if only authorized participants take part in the recovery phase, the ideal situation is that $t_j - 1$ participants want to recover the secret image. In our scheme, only if not less than t_j participants want to recover the secret will the dealer broadcast the corresponding key, r_j. By Theorem 3, we know that no participant can calculate her/his share of point pairs before broadcasting the key. Thus, less than t_j authorized participants cannot calculate their share point pairs, which means they cannot recover the secret image. Similarly, participants whose numbers are less than t_j cannot recover their share point pairs that correspond to threshold t_k $(k \in \{1, 2, \ldots, j-1, j+1, \ldots, N\})$ because only the dealer can broadcast the current key, r_j.

(2) Malicious and authorized participants both take part in the recovery phase.

Assuming that the current threshold is $t_j$and that $t_j \in \{t_1, t_2, \ldots, t_N\}$ if malicious and authorized participants both take part in the recovery phase, the ideal situation is that $k (1 \leq k \leq n - t_j + 1)$ malicious participants and $t_j - 1$ authorized participants want to recover the secret image. The malicious participants can disguise themselves as authorized participants. Without loss of generality, it is assumed that both malicious participants $\{M_1, M_2, \ldots, M_k\}$ and authorized participants $\{P_1, P_2, \ldots, P_{t_j-1}\}$ take part in the recovery phase. After broadcasting key r_j, authorized participants $\{P_1, P_2, \ldots, P_{t_j-1}\}$ can calculate their share point pairs, i.e., $(f(r_j, s_1), y_1^j)$, $(f(r_j, s_2)^j, y_2^j)$, and $(f(r_j, s_{t_j-1})^j, y_{t_j-1}^j)$, where $s_1, s_2, \ldots, s_{t_j-1}$ is their identification. According to feature (3) of the two-variable one-way function $f(r, s)$ presented in Section 3.1, malicious participants $\{M_1, M_2, \ldots, M_k\}$ cannot calculate their share point pairs without legal identification. Only if recovery have t_j share point pairs or more can the polynomial of degree $t_j - 1$, where secret data can be hidden, be recovered by the Lagrange interpolation formula. Thus, the secret image cannot be recovered in this situation.

To summarize, the secret image cannot be recovered when the number of authorized participants is less than the current threshold. □

Table 5. Comparison of related secret image sharing schemes.

Functionality	Yang et al. [22] 2007	Lin et al. [7] 2010	Ulutas et al. [26] 2013	Yuan et al. [31] 2016	Guo et al. [14] 2018	Liu et al. [33] 2019	Ours
Threshold	$\{t, n\}$	$\{t, n\}$	$\{t, n\}$	$\{t_1, t_2, \ldots, t_N\}$ $(t_{i+1} - t_i \leq t_1)$	$\{t, n\}$	$\{t', t, t''\}$	$\{t_1, t_2, \ldots, t_N\}$
Threshold changeability	No	No	No	Yes	No	Yes	Yes
Collusion Attack Resistance	Yes	Yes	Yes	Yes	Yes	No	Yes
Number of recovering polynomials	1	1	1	$\frac{N+1}{2}$	–	1	1
Meaningful stego image	Yes	Yes	Yes	Yes	Yes	No	Yes
Quality of stego images	46.0 dB	48.36 dB	52.79 dB	46.02 dB	48.0 dB	–	46.65 dB
Lossless secret image	Yes	Yes	Yes	Yes	Yes	No	Yes
Lossless cover image	No	Yes	Yes	No	No	No	Yes
Maximum capacity (pixels)	$\frac{H \times W}{4}$	$\frac{(t-1) \times H \times W}{3}$	$\frac{(t-2) \times H \times W}{4}$	$\frac{t_N \times H \times W}{N \times \lceil log_m 255 \rceil}$	$\frac{H \times W}{\lceil log_m 255 \rceil}$	–	*

– The scheme does not have this function; * The maximum capacity of our scheme is $\frac{t_1(t_1-1) \times (H_c \times W_c - \left\lceil \left\lceil \frac{log_m H_c \times W_c}{t_1} \right\rceil \times \frac{\lceil log_2 m \rceil}{w} \right\rceil)}{\lceil N(t_1-1)+t_1 \rceil \times \lceil log_m 255 \rceil}$.

Theorem 5. *Even if attackers steal the dealer's keys, they cannot recover the secret image without being able to identify the legal participants.*

Proof. Let us assume that the threshold is $t_j \in \{t_1, t_2, \ldots, t_N\}$ and that the attackers want to recover participant P_i's $(1 \leq i \leq n)$ first share point pair (x_i^j, y_i^j). According to feature (3) of the two-variable one-way function, i.e., $f(r, s)$, presented in Section 3.1, attackers cannot calculate $(f(r_j, s_i), y_i^j)$ without participant P_i's identification, s_i. Similarly, attackers cannot recover the rest of P_i's share point pairs and they cannot recover any participant's share point pairs without her/his identification.

In view of Theorem 4, attackers can recover the secret image only if they can obtain no less than t_j participants' share point pairs. Thus, attackers cannot recover the secret image without legal participants' identification even if they steal the dealer's keys. □

Theorem 6. *Even if attackers obtain the keys and all of the share point pairs used to recover the secret image, they cannot calculate any participant's identification.*

Proof. Assume that the current threshold is t_j and that the attackers want to calculate participant P_i's $(1 \leq i \leq n)$ identification, s_i, from key r_j and P_i's first share point (x_i^j, y_i^j). According to feature (5) of the two-variable one-way function, $f(r, s)$, presented in Section 3.1, attackers cannot calculate s_i from r_j and x_i^j. Similarly, attackers cannot obtain s_i from the rest of P_i's share point pairs and they cannot calculate any participant's identification. □

By Theorem 6, we know that attackers cannot obtain any participant's identification even if they obtain the keys and all share point pairs. Thus, participant's identification can be reused in subsequent secret image sharing procedures, which can improve the efficiency of our scheme.

6. Conclusions

This paper proposes a novel threshold-changeable secret image sharing scheme. The experiment shows that this scheme can produce high-quality stego images and can recover the cover image with loss. The theoretical analysis proved that our scheme can resist historical secret share attacks and collusion attacks, which means our scheme can adjust the threshold and recover the secret image securely. Thus, our scheme is safe and easy to use.

In the future, we will improve the stego image's quality by following these two aspects. One is to improve the share generation mechanism, which can reduce the amount of share; on the other hand, we will also design a high-capacity image hiding algorithm based on pixel prediction and compression sensing.

Author Contributions: conceptualization, G.W. and L.Y.; data curation, Q.W.; investigation, M.W.; methodology, M.W.; project administration, L.Y.; resources, Y.Y.; software, M.W.; supervision, G.W. and L.Y.; visualization, G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Zhejiang Province Natural Science Foundation under grant LY 19F020039 and by the open fund of Anhui Provincial Key Laboratory of Network and Information Security under grant AHNIS2019004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, C.; Jia, J.; Jie, Y.; Liu, C.Z.; Choo, K.R. Enabling Secure Cross-Modal Retrieval Over Encrypted Heterogeneous IoT Databases With Collective Matrix Factorization. *IEEE Internet Things J.* **2020**, *7*, 3104–3113, doi:10.1109/JIOT.2020.2964412.
2. Guo, C.; Jiang, X.; Choo, K.K.R.; Jie, Y. R-Dedup: Secure client-side deduplication for encrypted data without involving a third-party entity. *J. Netw. Comput. Appl.* **2020**, *162*, 102664, doi:10.1016/j.jnca.2020.102664.
3. Guo, C.; Jia, J.; Choo, K.K.R.; Jie, Y. Privacy-preserving image search (PPIS): Secure classification and searching using convolutional neural network over large-scale encrypted medical images. *Comput. Secur.* **2020**, *99*, 102021, doi:10.1016/j.cose.2020.102021.
4. Shamir, A. How to Share a Secret. *Commun. ACM* **1979**, *22*, 612–613, doi:10.1145/359168.359176.
5. Blakley, G.R. Safeguarding cryptographic keys. In Proceedings of the 1979 International Workshop on Managing Requirements Knowledge (MARK), New York, NY, USA, 4–7 June 1979; pp. 313–318.
6. Naor, M.; Shamir, A. Visual Cryptography. *Lect. Notes Comput. Sci.* **1994**, *950*, 1–12.
7. Lin, C.C.; Tsai, W.H. Secret image sharing with steganography and authentication. *J. Syst. Softw.* **2004**, *73*, 405–414, doi:10.1016/s0164-1212(03)00239-5.
8. Lin, P.Y.; Chan, C.S. Invertible secret image sharing with steganography. *Pattern Recognit. Lett.* **2010**, *31*, 1887–1893, doi:10.1016/j. patrec. 2010.01.019.
9. Guo, C.; Chang, C.C.; Qin, C. A multi-threshold secret image sharing scheme based on MSP. *Pattern Recognit. Lett.* **2012**, *33*, 1594–1600, doi:10.1016/j. patrec. 2012.04.010.
10. Guo, C.; Zhang, H.; Song, Q.; Li, M. A multi-threshold secret image sharing scheme based on the generalized Chinese reminder theorem. *Multimed. Tools Appl.* **2015**, *75*, 11577–11594, doi:10.1007/s11042-015-2885-x.
11. Liu, Y.N.; Zhong, Q.; Xie, M.; Chen, Z.B. A novel multiple-level secret image sharing scheme. *Multimed. Tools Appl.* **2018**, *77*, 6017–6031, doi:10.1007/s11042-017-4512-5.
12. Zarepour-Ahmadabadi, J.; Shiri Ahmadabadi, M.; Latif, A. An adaptive secret image sharing with a new bitwise steganographic property. *Inf. Sci.* **2016**, *369*, 467–480, doi:10.1016/j.ins.2016.07.001.
13. Yuan, L.; Li, M.; Guo, C.; Choo, K.K.R.; Ren, Y. Novel Threshold Changeable Secret Sharing Schemes Based on Polynomial Interpolation. *PLoS ONE* **2016**, *11*, e0165512, doi:10.1371/journal.pone.0165512.
14. Guo, C.; Zhang, H.; Fu, Z.; Feng, B.; Li, M. A novel proactive secret image sharing scheme based on LISS. *Multimed. Tools Appl.* **2017**, *77*, 19569–19590, doi:10.1007/s11042-017-5412-4.
15. Laih, C.S.; Harn, L.; Lee, J.Y.; Hwang, T. Dynamic Threshold Scheme Based on the Definition of Cross-Product in an N-Dimensional Linear Space. In *Advances in Cryptology—CRYPTO' 89 Proceedings*; Brassard, G., Ed.; Springer: New York, NY, USA, 1990; pp. 286–298.
16. Desmedt, Y.; Jajodia, S. *Redistributing Secret Shares to New Access Structures and Its Applications*; Technical Report; Citeseer: State College, PA, USA, 1997.
17. Martin, K.; Pieprzyk, J.; Safavi Naini, R.; Wang, H. Changing Thresholds in the Absence of Secure Channels. *Aust. Comput. J.* **1999**, *31*, 34–43.
18. Barwick, S.G.; Jackson, W.; Martin, K.M. Updating the parameters of a threshold scheme by minimal broadcast. *IEEE Trans. Inf. Theory* **2005**, *51*, 620–633, doi:10.1109/TIT.2004.840857.
19. Zhang, Z.; Chee, Y.M.; Ling, S.; Liu, M.; Wang, H. Threshold changeable secret sharing schemes revisited. *Theor. Comput. Sci.* **2012**, *418*, 106–115, doi:10.1016/j. tcs. 2011.09.027.
20. Pilaram, H.; Eghlidos, T. A lattice-based changeable threshold multi-secret sharing scheme and its application to threshold cryptography. *Sci. Iran.* **2017**, *24*, 1448–1457, doi:10.24200/sci.2017.4126.
21. Jia, X.; Wang, D.; Nie, D.; Luo, X.; Sun, J.Z. A new threshold changeable secret sharing scheme based on the Chinese Remainder Theorem. *Inf. Sci.* **2019**, *473*, 13–30, doi:10.1016/j.ins.2018.09.024.
22. Yang, C.N.; Chen, T.S.; Yu, K.H.; Wang, C.C. Improvements of image sharing with steganography and authentication. *J. Syst. Softw.* **2007**, *80*, 1070–1076, doi:10.1016/j.jss.2006.11.022.
23. Guo, C.; Chang, C.C.; Qin, C. A hierarchical threshold secret image sharing. *Pattern Recognit. Lett.* **2012**, *33*, 83–91, doi:10.1016/j.patrec.2011.09.030.
24. Pakniat, N.; Noroozi, M.; Eslami, Z. Secret image sharing scheme with hierarchical threshold access structure. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1093–1101, doi:10.1016/j.jvcir.2014.03.004.
25. Wu, Z.; Liu, Y.; Jia, X. A novel hierarchical secret image sharing scheme with multi-group joint management. *Mathematics* **2020**, *8*, 448, doi:10.3390/math8030448.
26. Ulutas, M.; Ulutas, G.; Nabiyev, V.V. Invertible secret image sharing for gray level and dithered cover images. *J. Syst. Softw.* **2013**, *86*, 485–500, doi:10.1016/j.jss.2012.09.027.
27. Chen, C.C.; Chen, J.L. A new Boolean-based multiple secret image sharing scheme to share different sized secret images. *J. Inf. Secur. Appl.* **2017**, *33*, 45–54, doi:10.1016/j. jisa. 2017.01.006.
28. Nag, A.; Singh, J.P.; Singh, A.K. An efficient Boolean based multi-secret image sharing scheme. *Multimed. Tools Appl.* **2019**, *79*, 16219–16243.
29. Deshmukh, M.; Nain, N.; Ahmed, M. An (n, n)-multi secret image sharing scheme using boolean XOR and modular arithmetic. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 27–29 March 2016; pp. 690–697.

30. Azza, A.; Lian, S. Multi-secret image sharing based on elementary cellular automata with steganography. *Multimed. Tools Appl.* **2020**, *79*, 21241–21264.
31. Yuan, L.; Li, M.; Guo, C.; Hu, W.; Luo, X. Secret Image Sharing Scheme with Threshold Changeable Capability. *Math. Probl. Eng.* **2016**, *2016*, 1–11, doi:10.1155/2016/9576074.
32. Thien, C.C.; Lin, J.C. Secret image sharing. *Comput. Graph.* **2002**, *26*, 765–770, doi:10.1016/S0097-8493(02)00131-0.
33. Liu, Y.X.; Yang, C.N.; Wu, C.M.; Sun, Q.D.; Bi, W. Threshold changeable secret image sharing scheme based on interpolation polynomial. *Multimed. Tools Appl.* **2019**, *78*, 18653–18667, doi:10.1007/s11042-019-7205-4.
34. Chien, H.Y.; Jan, J.K.; Tseng, Y.M. A practical (t, n) multi-secret sharing scheme. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2000**, *E83-A*, 2762–2765.
35. He, J.; Dawson, E. Multi secret-sharing scheme based on one-way function. *Electron. Lett.* **1995**, *31*, 93–95.
36. Goldwasser, S.; Micali, S.; Rivest, R.L. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.* **1988**, *17*, 281–308.
37. Naor, M.; Yung, M. Universal One-Way Hash Functions and their Cryptographic Applications. In Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, 15–17 May 1989; pp. 33–43.
38. Waring, E. Vii. problems concerning interpolations. *Philos. Trans. R. Soc. Lond.* **1779**, 59–67, doi:10.1098/rstl.1779.0008.
39. Kyriakopoulos, K.; Parish, D.J. A live system for wavelet compression of high speed computer network measurements. In Proceedings of the International Conference on Passive and Active Network Measurement, Louvain-la-Neuve, Belgium, 5–6 April 2007; pp. 241–244.