Chapter 1

Introduction

1.1 Motivation

Computer network is very popular nowadays. They are useful in both communication and entertainment. People can transmit information through computer network easily. However, because of the open environment of network, hackers may steal or destruct the information transmitted in the network. Losing some data such as family photos or chatting records are usually of no big deal, but some other information, for example, the reports of the newshound, the financial budget of a company, or the military secrets of a country, is so important that losing them will cost much money, time, or even lives.

Due to the risk of being hacked, people develop lots of cryptography method to encrypt the important data. For example, DES (Data Encryption Standard), AES (Advanced Encryption Standard), and RSA are all famous cryptography methods still being used nowadays.

The methods mentioned above are based on the so-called keys: namely, only the person who has the key can extract the secret data. Besides using the keys, the other way to protect data is using the "sharing" technology to share data. Shamir [1] proposed a sharing method. The user can use his polynomial method to share the data among n shadows. The shadows may then be distributed to n different persons. If some of the users need the secret data, he or she must collect at least r shadows ($r \le n$) to recover the secret data. The sharing method assures that collecting r or more than r

shadows can reconstruct the secret data; however, if there are only *r*-1 or less shadows, then these shadows reveal nothing about the secret information.

The method proposed by Shamir can also be used for image sharing. The gray values of the pixels are treated like secret data. Using Shamir's method can share the secret image among *n* shadows, and collecting *r* of the *n* shadows can recover the secret image. However, the *n* shadows used a lot of storage space because each one is of the same size as the secret image is. Notably, a lossy compressed method called Vector Quantization (VQ) is a convenient compression method [2]. By using VQ, although the image sacrifices its quality, the data size of the shadows can be reduced to smaller ones. Thus, before sharing, each user may apply VQ to his secret image. Then the code indices are shared (instead of the gray values). This will cause the sharing worthier because the data amount is reduced.

Besides traditional sharing, people may also need some progressive sharing methods to achieve some goals. For example, in a company, we may desire that the general manager can get a good quality version of the secret image; while a local manager can receive a medium quality version of the secret image, and an ordinary staff can only get a low quality one. The progressive image sharing method proposed in this thesis can achieve the purpose. In our progressive sharing schemes, collecting more shadows will make the quality of the recovered secret image better. In that scheme, the lossless secret image can be generated after all of the shadows are received. With this method, we can allocate more shadows to the members of higher position and thus let them have the power to reconstruct better-quality images.

Although the sharing technique can reduce the risk that the secret image being hacked, there is still some chance that the shadows are damaged or modified. If one or more of the shadows are damaged, the secret image cannot be recovered perfectly. In this condition, the user can only ask for another non-damaged shadow. However, if all shadows are damaged, then no method can recover the secret image. Thus we proposed an error correction method that the user can use the generated auxiliary image to correct the damaged secret image. The generated auxiliary image not only can improve the damaged image, but also reveals nothing about the secret image if the auxiliary image is used alone without being together with the received (broken) shadows.

1.2 Related Works

1.2.1 Literatures about Image Sharing

Shamir [1] first proposed a sharing method using polynomial shown below:

$$q(x) = (a_0 + a_1 x + a_2 x^2 ... + a_{r-1} x^{r-1}) \bmod p . \tag{1.1}$$

Here, a_0 is the secret data (a number), and p is a given prime number. The coefficients a_1, a_2, ..., a_{r-1} are randomly generated from the set of the integers between 0 and $p-1$. By Equation (1.1), the user can calculate $q(1) \sim q(n)$, which are the shadow numbers. After collecting at least r shadow numbers, the secret number a_0 can be calculated using the Lagrange's interpolation. After Shamir's proposal, many researches of secret sharing were proposed [3-6]. Thien and Lin [7] proposed a modified method that didn't use the randomly generated coefficients $a_1 \sim a_{r-1}$. Instead, the coefficients a_0, a_1, a_2, ..., a_{r-1} are all secret data. Thus, the n shadows all became to $1/r$ in size of the original secret image. This benefit lowers the storage space and transmission time. Wu, Thien, and Lin [8] used the sharing method in [7] and S-E table to reduce further the size of stego images.

Besides the sharing using polynomial, Tsai *et al.* used the exclusive-OR operation to share the secret [9]. Visual cryptography [10-13] is a convenient method to share the image. By adjusting the luminance of the extended pixel in the shadow transparency, the secret data is shared among shadow transparencies, and nothing will reveal from each transparent sheet alone. The user can stack two or more transparencies to see the secret data visually without the need of complicated computation. Visual cryptography is hard used to deal with color images; instead, its application is usually restricted to monochrome images.

The progressive image sharing approach is another choice to share the secret image. Manohar and Tilton [14] proposed a lossless image sharing method. However, the incoming shadows must be in order; otherwise, the secret image can not be recovered. Hung [15] developed a progressive image sharing method on frequency domain. The different-quality secret images can be recovered by different numbers of any shadows, but it cannot recover image losslessly, and the threshold is hard to adjust.

One of the topics of this thesis — vector quantization (VQ) [2] — provides a simple method to compress the image by using the codebooks and the code indices. Many researchers developed the method to speed up the generation of the codebooks and code indices [16-19]. Chang and Hwang [20] achieved the goal of (r, n) threshold scheme by sharing the codebook. Chen and Chang [21] used the host images as codebooks, and hide the code indices of the secret image into these host images. Collecting all of the shadows can reconstruct the secret image. However, their method does not provide the (r, n) threshold scheme because if one of the shadows is lost then the reconstruction of the secret image becomes impossible.

1.2.2 Literatures about Image Hiding

The secret is often hidden into some images. The images where the secret will be hidden are called host images. Then, the images which already contain the information about the secret are called shadow images. There are many researches about image hiding to reduce the difference between the host images and shadow images. The simplest hiding method is the least significant bit (LSB) method. It replaces the least *m* bits of the host images by the secret data (the value of *m* depends on the size of the host image and secret data). To optimize the quality of shadow images, Wang *et al*. [22] used Genetic Algorithm to improve the LSB substitution method. Thien and Lin [23] proposed another image hiding method based on modulus operation by modifying LSB method. Both the theoretic results and the vision quality are better than the simple LSB substitution method and the GA-improved method. Wang *et al*. [24] also designed a moderately significant bit (MSB) hiding method by the use of optimal substitution process and local pixel adjustment. This provides another choice to hide the secret data.

1.2.3 Literatures about Fake Detection and Error Correction

Though the (r, n) threshold scheme provides a fault-tolerant method to share the secret, the risk that the number of the damaged shadows is more than *r* (because some shadows are modified by hackers) still exists. Tompa and Woll [25] showed how to cheat in the threshold scheme. Thus many researches provide the methods against the cheater. Wu and Wu [26] and Hwang *et al*. [27] used one-way hash function to detect and identify the cheat. Chang and Hwang [28] proposed a method using quadratic residues instead of the hash functions. Lee and Won [29] used watermarking to correct the alterations. Hung [15] provided a method using additional shadow images to detect the damaged shadow and correct the damaged recovered image.

1.3 Overview of the Proposed Methods

In this section we will briefly describe the methods proposed in this thesis.

Firstly, the image sharing and hiding method of VQ-style shown in Fig. 1.1 will be introduced. By using the host images as codebooks, and hiding the code indices and the mixed information of the codebooks into the host images, we provide a method of (r, n) threshold scheme of VQ-style to share the secret image. Because the hidden information is of small amount (only the code indices and the mixed information of the codebooks), the impact to the host images is small, so that the shadow images look like original host images after hiding. The recovered quality of the secret image is not bad, too (the same as that of the original VQ).

Secondly, the lossless progressive image sharing by VQ will be shown. By calculating the code indices of the secret image and the difference between the recovered image (recovered by the code indices, thus this image is lossy) and the original image, the secret image can be recovered by using the shadow images which contain the information of code indices and the differences. The more shadows the user gets, the better the quality of the recovered secret image. Besides, the reconstructed image depends only on the number of the shadows. Thus the user does not need to worry about the order or which shadows he/she gets (the user only needs to care about how many shadows he/she receives). Finally, after collecting all of the shadows, the secret image can be recovered in a loss-free manner.

In next topic, we will propose an error correction method using search-order coding (SOC), as is shown in the flowchart of Fig. 1.2. The secret image is first processed by the technique of SOC, which will generate an image (we call it SOC-image). Then, the secret image can be shared by any sharing method (whether it is polynomial approach or not). Note that no one can get the secret image unless using reconstruction from the shadows. If one or more shadows are damaged, the secret image cannot be recovered perfectly. However, by using the SOC-image, the quality of the damaged secret image can be improved. Though the SOC-image cannot repair the damaged image completely to a perfect one, it indeed can meliorate the damaged image. Although it has the ability of the repairing, the SOC-image alone reveals nothing about the secret image. So it is much safer than just backing up the original secret image.

Finally, we will show some applications of the SOC-image. The SOC-image can be not only applied to normal images (formed of the gray values of the pixels), but also used in VQ file (formed of the code indices). The experimental results will show that it is indeed useful and can correct the code indices in VQ. The advantages of the SOC-image will be introduced, too. The security and the compression of the SOC-image will both be addressed. The experimental results will show that the SOC-image is useful in progressive image sharing, too.

1.4 Dissertation Organization

This thesis includes the image sharing part and the image recovery part. In Chapter 2, two methods of image sharing of VQ-style are proposed. One is the fault-tolerant sharing and the other is the progressive sharing. The error correction by SOC is introduced in Chapter 3. Chapter 4 combines the SOC with VQ, followed by introducing the applications of SOC recovery. Finally, the conclusion and future works are discussed in Chapter 5.

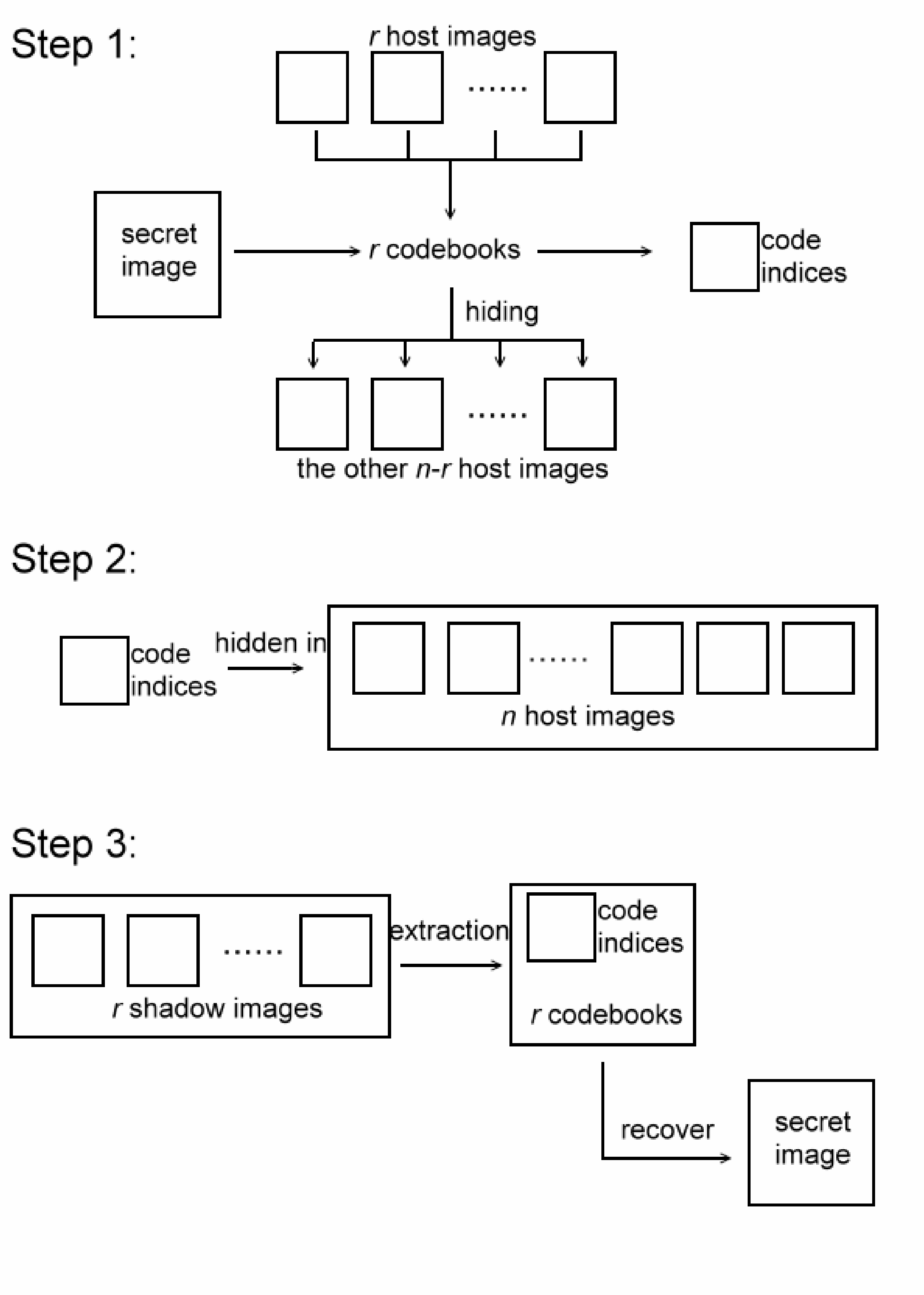

Fig. 1.1 The approach of the (r, n) sharing.

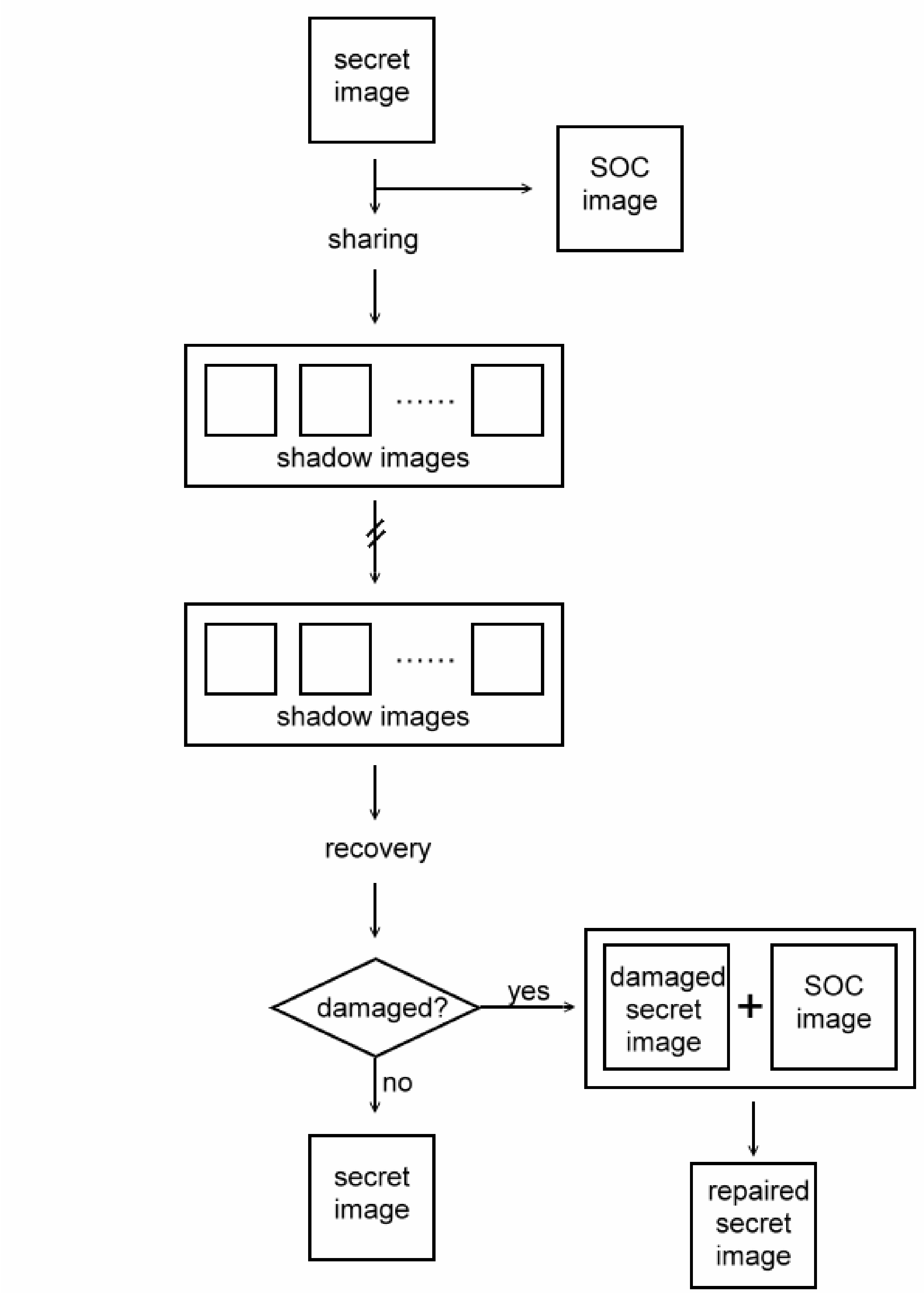

Fig. 1.2 The flowchart of the SOC recovery technique.