Second International Symposium on Computer Vision and the Internet (VisionNet’15)

A Secure Verifiable Scheme for Secret Image Sharing

Angel Rose A\textsuperscript{a}, Sabu M Thampi\textsuperscript{b}

\textsuperscript{a,b}IIITM-K, Trivandrum, Kerala, India

Abstract

Verifiable Secret Image Sharing has become an important field of study in modern cryptography. As the era demands the need of security and verifiability to resist cheating scenario, a new secret image sharing scheme identifying the existence of cheater is introduced and analyzed in this paper. A method to ensure integrity of secret image prior to its recovery is proposed. An \(n \times n\) secret image and \(n \times n\) verification image are used to generate shares which are embedded into a cover image for transmission. Structural similarity and mean square error measure of reconstructed verification image with original verification image verifies the coherence of the secret. Computational cost of this method is low which makes it suitable for covert message communication and sharing of scanned documents.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of organizing committee of the Second International Symposium on Computer Vision and the Internet (VisionNet’15)

Keywords: Verifiable Secret Image Sharing ; Steganography
1. Introduction

Secret Image sharing refers to a cryptographic technique in which secret image is divided into a number of share images with or without modification and the secret image can be retrieved by combining all or predefined collection of share images. In any basic secret image sharing protocol there are two entities - Dealer and Participants. Dealer is an entity who handles the task of share construction and distributes these shares to various distributed participants. Participants are those entities who receive the share images from dealer and participate in secret recovery process. A dishonest entity may tamper the share, thereby making the secret reconstruction ambiguous. This poses a security breach and integrity of secret image is lost, which requires the need of extending ordinary secret image sharing scheme.

Security of secret sharing scheme is defined under the assumption that all the participants and the dealer involved are honest. If any one of the entities acts as cheater and tries to damage the shares, secret recovery is compromised. Hence, there is need for Verifiable Secret Image Sharing.

Fig 1: Share Construction and Distribution

In a Verifiable Secret Image Sharing Scheme, two types of attack can occur. Either dealer or participants can act as dishonest. Dealer may distribute incorrect share images to some of the participants or a participant may supply an incorrect share image, when a group of participants are trying to recover the original secret image. Dealer can be a sender who wishes to share secret image or a trusted third party. Sometimes dealer plays the role as cheater and it may intentionally or deliberately create wrong shares to distribute, thus hinder secret sharing. But rarely this kind of scenario can occur; in most cases this won’t happen. While designing a secret sharing scheme, one should be aware of these attacks. As visual secret sharing and its application continue to be studied extensively, the cheating problem has become an important issue. The cheating process stacks damage share and genuine share together to retrieve secret. Verifiability allows involved participants to detect cheaters during secret retrieval process.

Many existing schemes make use of additional authentication shares for verifying the integrity of shares\textsuperscript{16}, or keeping additional transparencies\textsuperscript{17} for verification purpose. Both involve additional overhead. Wang et.al\textsuperscript{5} proposed a (2,2) secret sharing scheme\textsuperscript{1}, which verifies the integrity of share images with the help of watermark image. At the same time, it employs an image chaotic technique called torus automorphism\textsuperscript{22} during share construction phase. Torus automorphism is much time consuming and timestamp varying in accordance with characteristics of input image. The security of Wang’s scheme is primarily defined in terms of torus automorphism and the shares produced are of meaningless or not realistic in nature.

Defining characteristics of share image is trivial i.e., shares should not reveal any information or hint about the secret image and also be less susceptible to malicious attacks during transit. In our scheme, we assume that a dealer is a trusted entity. Interactive participants may or may not act as cheater. The proposed scheme verifies the integrity of secret image with the help of verification image. Shares produced are realistic images. Low computational requirements and security features makes this scheme suitable for real time applications such as covert message...
communication over internet and sharing of scanned documents. The scheme verifies the integrity of secret image prior to its recovery based on the structural similarity and mean square error measure of verification image with the reconstructed one. In the rest of this paper, section 2 reviews the literature. Section 3 introduces our verifiable secret image sharing scheme in detail. Section 4 discusses the experimental results. Finally, section 5 concludes the paper.

2. Literature Review

In 1979 Blakley\(^1\) and Shamir\(^2\) first proposed the concept of secret sharing through a \((t,n)\) threshold scheme. i.e. \(t\) out of \(n\) shares are required to reconstruct the secret. As secret image sharing and its application continued to be studied extensively, \((n, n)\) and \((2, 2)\) scheme got more attention among other schemes.

Naor and Shamir\(^3\) were the first to propose a \((t, n)\) visual cryptography technique that involves dividing an image into \(n\) different shares. The secret image could be reconstructed by stacking at least \(t\) shares among the \(n\) share images and stacking of \(< t\) shares never allow the decryption to be successful. Many visual secret image sharing methods have been proposed in the past few years, few of the schemes\(^4\) are designed to allow a particular subgroup of defined participants needed to decrypt the secret, and some\(^\text{18, 19}\) designed to allow any sub-group of \(n\) participants, and these are the most popular. Among them, \((2, n)\) and \((n, n)\) schemes are widely used.

Lukac and Plataniotis proposed bit level-based image secret-sharing\(^12\), where a gray or color image is decomposed into several binary bit-planes. Each bit-plane is treated as a binary secret image that is processed using Visual Cryptography (VC) encryption to form two share images. After that, two gray or color-share images are generated by composing all the binary share images through stacking these bit-planes. This scheme lacks verification ability and it’s difficult to manage noise like shares. Pixel expansion is also a drawback.

Tsai et al. proposed an image-sharing scheme\(^9\) that combines image-hiding and VC. In this scheme, secrets are divided into multiple parts that are hidden in the bit-planes of a set of cover images to form stego-images. The aim of this scheme is to prevent anyone who processes only one stego-image from gaining information about the secret. As VC and VC-based applications continue to be studied extensively, the cheating problem-stacking fake and genuine share together to reveal the secret-that exists in VC has been highlighted, and preventing cheating has become an important issue. In 2007, Horng et al.\(^13\) proposed a method to identify cheaters with the cost of additional authentication shares which authenticate the integrity of shares prior to stacking them. This method is not computationally effective as well as noise like shares bring extra cost of managing shares. The shares produced are larger than secret image.

Most of the verifiable secret-sharing schemes proposed allow participants to verify only their received shares instead of reconstructed secret image, but many have the issues of bad recovering quality, pixel expansion (share size), computational complexity, security and accuracy. Wang et al.\(^5\) proposed a visual sharing method with verification ability which first applies few equations to encode a watermark image and a secret image into two non-expanded sharing image. This is followed by scrambling of two sharing images using torus automorphism. In this scheme, it is difficult to manage meaningless shares and Torus automorphism consumes time to scramble images.

Recently in 2013, Hao-Kuan Tso proposed a scheme\(^7\) in which gray scale image is converted to bit plane images and each bit plane image along with seed and verification image encoded image is produced. This scheme is capable of producing meaningful shares as well as solves pixel expansion problem, however meaningless are not realistic images and hence still attracts attacker’s attention. In this scheme, both seed and verification image need to be kept secret. Another issue is one of the shared images cannot reveal any information of the original images. However, whenever enough shared images are obtained, the original information would be revealed progressively.

Our scheme addresses the issues identified in the existing schemes. The proposed scheme produces realistic shares which follows same size correlation with the secret image. This secure scheme verifies the integrity of shares prior to its recovery.
3. Proposed Scheme

The proposed scheme consists of three phases: Share Construction Phase, Information Hiding Phase and Revealing and Verification Phase as shown in Fig.2.

An \( n \times n \) binary verification image and \( n \times n \) binary secret image are input to the share construction phase and two share images share1 and share 2 are produced. Share construction is carried out at bit-level using Eqn. (1) and Eqn. (2). Details of share construction algorithms described in the next section. Produced shares do not reveal any information regarding both secret as well as verification image. During Information Hiding phase, generated shares are separately hidden inside two user selected \( n \times n \) grayscale cover images and stego-share images are produced. Now the resulted shares are meaningful. These shares are sent to various participants.

\[
S_{ij}^A = \left\lfloor \frac{(O_{ij} \times 2 + V_{ij} + 1) \mod 4}{2} \right\rfloor \quad (1)
\]
\[
S_{ij}^B = \left\lfloor \frac{(O_{ij} \times 2 + V_{ij} + 1) \mod 2}{2} \right\rfloor \quad (2)
\]

However, it is observed that, produced shares are informative i.e. it reveals some information regarding secret and verification image. So, in order to scramble the pixel information further, we followed Arnold transformation technique and finally, share \( S^A \) and share \( S^B \) are produced.

In order to recover the secret image in a lossless manner, interactive participants collect the share images and reconstruct the verification image first, followed by secret image recovery. The verification image helps participants to identify whether the collected shares are damaged. The detailed description of each phase is given in the following subsections.

A. Share Construction Phase

Share construction phase consist of 4 steps as illustrated in flow chart Fig.3. Dealer initiates the activities of share construction utilizing the equations rather than merely depending on any code book. The resulted shares are then given to next phase. The original secret image \( O \) and Verification image \( V \) are input to this phase. First of all two shares are constructed using Eqn. (1) and (2).

Fig.2: The proposed scheme
Algorithm 1 Share construction algorithm

Input: An H×W Original secret image \( O = (O_{ij}) \),
an H × W Verification image \( V = (V_{ij}) \),
Where \( i = 0, 1, \cdots, H - 1 \) and \( j = 0, 1, \cdots, W - 1 \).
Output: Two \( H \times W \) noise-like share images \( S^A = (S^A_{ij}) \) and \( S^B = (S^B_{ij}) \) Where \( i = 0, 1, \cdots, H - 1 \) and \( j = 0, 1, \cdots, W - 1 \).

begin
1: Set \( i = 0 \) and \( j = 0 \),
2: Read pixel \( O_{ij} \) from the secret image \( O \) and read verification pixel \( V_{ij} \) from the verification image \( V \).
3: repeat
   4: Obtain pixel of share \( S^A \), utilizing Eqn.(1)
   5: Find the value for pixel of share \( S^B \), using Eqn.(2)
   6: until all pixels are processed
7: Apply Arnold Transformation to permute them to become random share images \( S^A \) and \( S^B \), sized \( H \times W \).
end

Arnold transformation serves two purposes. One is security of the scheme improved and other is to make shares suitable for applying BPCS Steganography during Information Hiding Phase.

B. Information Hiding Phase

Two noise-like random shares are hidden inside user-selected camouflage images by applying steganography technique. Bit-Plane Complexity Steganography (BPCS) Technique^{13} is used for this purpose. In this scheme, cover image is divided into bit-plane images and complex blocks are found based on black and white border length. The complex block of bit plane images is replaced with random binary patterns. BPCS Steganography involves two processes, embedding and extraction. During BPCS embedding, complex regions in each bit-plane of cover image are replaced with random share blocks i.e. Resulted stego share images are meaningful and realistic picture of higher quality. The overall block diagram of BPCS embedding steps is shown in Fig.5. The secret image can be either share1 or share2. Once complex blocks are replaced, bit plane images are combined back to form stego share images (Fig.4). These stego shares are then sent to the participants.
Interactive participants do the reverse process to recover the shares $S^A$ and $S^B$ once they receive stego share images. This process is called as BPCS Extraction and the steps are illustrated in Fig. 6 and finally random shares are recovered. The stego Image can be of either stego-share1 or stego-share2.

C. Revealing and Verification Phase

Revealing and verification phase is the final and trivial phase of the proposed method. The proposed revealing and verifying phase allows the defined participants to extract the embedded verification image and reconstruct the secret image from two random shares, $S^A$ and $S^B$. This operation is carried out in bit-level using Eqn. (3) and (4):

$$O'_{ij} = \left\lfloor \left( S^A_{ij} \times 2 + S^B_{ij} + 3 \right) \mod 4 \right\rfloor / 2$$

$$V'_{ij} = \left\lfloor \left( S^A_{ij} \times 2 + S^B_{ij} + 3 \right) \mod 2 \right\rfloor$$
The quality of the reconstructed secret image is same as that of the original secret image and has no distortion. Moreover, the procedure helps legitimate participants to verify the reconstructed secret image $O'$ based on the extracted verification image $V'$. Overall working of this phase is illustrated in Fig. 7.

**Algorithm 2**

Revealing algorithm

| Input: | Two $H \times W$ noise-like random share images $S^A = (S^A_{ij})$ and $S^B = (S^B_{ij})$ where $i = 0, 1, \cdots, H - 1$ and $j = 0, 1, \cdots, W - 1$. |
| Output: | An $H \times W$ reconstructed secret image $O' = (O'_{ij})$, an $H \times W$ reconstructed verification image $V' = (V'_{ij})$ |

begin
1: Perform inverse Arnold transformation and obtain the intermediate shares $S^A$ and $S^B$
2: Set $i = 0$ and $j = 0$, Read the pixel $S^A_{ij}$ of share $S^A$ and the pixel $S^B_{ij}$ of share $S^B$
3: repeat
4: Reconstruct pixel of secret image $O'_{ij}$ using Eqn.(3)
5: Reconstruct pixel of verification image $V'_{ij}$ using Eqn.(4)
6: until all pixels are processed
7: Determine the difference between the original verification image and the reconstructed verification image
end

The extracted verification image can be verified by using mean square error (MSE) and Structural Similarity Index value test. If the MSE value is equal to 0 or Structural Similarity Index value is equal to 1, the extracted verification image is same as the hidden one and there has been no cheating by the participant. Otherwise, the receiver can ask the dealer to distribute the entire share again.

4. Experiments

The proposed method is implemented in Matlab 2010 b. The results obtained are discussed. To demonstrate how the scheme successfully achieves general criteria such as security, accuracy, computational complexity, and pixel expansion, this section presents a set of experimental results. Experimental results on four general criteria are discussed in subsections 4.1, 4.2, 4.3 and 4.4.

To illustrate our scheme, we used a set of test images as shown in Fig. 8. The set contains $512 \times 512$ binary images: “Peppers,” “GoldHill,” and $512 \times 512$ grayscale images: “lena,” “baboon”. In the test set, “Peppers” serve as the secret image, while “GoldHill” serves as the verification image which are used to verify the reconstructed secret images. Cover images are “lena,” and “baboon”.

![Diagram](image-url)
The secret “peppers” and verification image “Goldhill” are input to share construction phase and the produced two shares are shown in Fig. 9(a) & 9(b).

Share1 was hidden inside “lena” cover image during BPCS embedding process and stego share 1 was produced as shown in Fig.9 (c). Similarly, share2 was hidden inside “baboon” image to form stego share2 as shown in Fig.9 (d). These are meaningful shares which are sent to participants. Later, original shares are extracted from these shares during BPCS extraction step. During Revealing and validation, we recovered the verification image first and the resulted image is same as the original verification image as shown in Fig.9 (e). Finally, secret image is recovered lossless with PSNR value ‘∞’ as shown in Fig.9 (f).

4.1 **Security analysis.** To guarantee that our scheme satisfies the security criterion i.e. it avoids leaking any information about the original secret image, Arnold transformation is used. It relocates the constructed share pixels after they have been generated. The sets of shares generated by our scheme are shown in Fig.9 (a) & 9(b). Also complex BPCS Steganography technique further hides the existence of these shares by producing quality stego images as in Fig.8 (c) & 9(d).

4.2 **Accuracy.** In our experiments, the peak signal-to-noise ratio (PSNR) is used to evaluate the quality of the reconstructed secret image with original secret.

The peak signal-to-noise ratio measures the quality by first calculating the mean squared error (MSE) and then dividing the maximum range of the data type by the MSE as shown in Eqn. (5).

\[
PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \quad (5)
\]

\[
MSE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} (Q_{xy} - Q'_{xy})^2 \quad (6)
\]

For an original gray scale image with a size of H × W, the corresponding MSE is defined in Eqn. (6). Basically, PSNR value should range from 30dB to 40dB if a scheme offers good visual quality. The PSNR value equals to ∞
denotes maximum quality. To determine the accuracy of our scheme, recovered secret image is compared with original image.

4.3 **Computational complexity.** According to the descriptions of our proposed scheme, the computational cost depends on only two operations: the sum operation and Arnold transformation in share construction phase. Clearly, the complexity of the sum operation is very low and has very little effect on the computational complexity of our scheme. The execution time of our scheme for “pepper” and “baboon” is shown in Table 1.

Table 1: Execution time

<table>
<thead>
<tr>
<th>Share 1</th>
<th>Share 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.756362 seconds</td>
<td>3.754314 seconds</td>
</tr>
</tbody>
</table>

4.4 **Pixel expansion.** During the share construction process, our proposed scheme generates two shares each with size of original secret. Table 2 shows the correlation between the original image size and the share size.

Table 2: Size Comparison

<table>
<thead>
<tr>
<th>Image</th>
<th>Binary images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the original image</td>
<td>2097152 bytes</td>
</tr>
<tr>
<td>Size of the share $S_A$</td>
<td>2097152 bytes</td>
</tr>
<tr>
<td>Size of the share $S_B$</td>
<td>2097152 bytes</td>
</tr>
</tbody>
</table>

4.5 **Evaluation of verifying.** The mean square error (MSE) and Structural Similarity value is used to identify whether cheating has occurred by comparing the content of an original verification image with that of an extracted one. If the MSE value is equal to zero or SSIM index value is equal to 1, there is no difference between images. If the MSE value is not equal to zero or SSIM index value is not equal to one, then reconstructed secret image $I'$ is unreliable. Fig.4.2 (e) & (f) show the extracted verification image and the reconstructed secret images and their verification results. Experiments were conducted using database 23. Results are given in Appendix A.

Experiments were conducted by partially damaging a share by adding black pixels (Black Damage), white pixels (White Damage), mixing with any images (Mix Damage), random pixels (Random Damage) etc; the corresponding MSE of the extracted verification image is not equal to “0” or SSIM Index value not equal to “1” for “Peppers” and “Baboon”. Details are shown in Table 3.

Table 3: Verification Result ("Share1 is partially damaged")

<table>
<thead>
<tr>
<th>White Damage</th>
<th>Black Damage</th>
<th>Mix Damage</th>
<th>Random Damage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.17 (MSE)</td>
<td>.07</td>
<td>.31</td>
<td>.12</td>
</tr>
<tr>
<td>.9901(SSIM index)</td>
<td>.996241</td>
<td>.964165</td>
<td>.99554</td>
</tr>
</tbody>
</table>

The result is graphically represented in Fig 10. It shows the verification result values of MSE value (>0) and SSIM index value (< 1) under various damage scenarios.

![Fig.10: Verification Result](image_url)
Proposed visual secret-sharing scheme for binary images satisfies the four general criteria required for visual secret sharing systems. Moreover, the computational complexity of our proposed scheme is very low, so it is suitable for real-time applications.

5. Conclusion

Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. A verifiable secret sharing scheme allows participants to be certain that no other participants are lying about the contents of their shares up to a reasonable probability of error. The scheme satisfies the basic features of secret sharing scheme and it solves the issues identified in Wang et.al scheme. Shares of secret are meaningful and hence don’t attract the attention of intruders.

The proposed scheme is capable of identifying whether cheater exists or not. The work can be extended to find out exact cheater identification and tampered location in case of damage to share occurs. Further, we can extend the scheme to (t, n) visual secret sharing scheme applicable for various image types.

Acknowledgment

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

Appendix A

<table>
<thead>
<tr>
<th>Secret Image</th>
<th>Verification Image</th>
<th>CoverImage 1</th>
<th>Cover Image 2</th>
<th>StegoShare 1</th>
<th>StegoShare2</th>
<th>Recovered Verification Image</th>
<th>Recovered Secret Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kodim01 16.7 KB</td>
<td>Kodim02 2.03KB</td>
<td>Kodim03 129 KB</td>
<td>Kodim04 143 KB</td>
<td>41.9282 dB</td>
<td>46.5717 dB</td>
<td>PSNR=∞ 16.7KB</td>
<td></td>
</tr>
<tr>
<td>Kodim03 8.89KB</td>
<td>Kodim04 6.12KB</td>
<td>Kodim07 134 KB</td>
<td>Kodim08 176 KB</td>
<td>42.6146 dB</td>
<td>47.4061 dB</td>
<td>PSNR=∞ 8.89KB</td>
<td></td>
</tr>
<tr>
<td>Kodim05 11.3 KB</td>
<td>Kodim06 10.5 KB</td>
<td>Kodim11 152 KB</td>
<td>Kodim12 133 KB</td>
<td>46.8763 dB</td>
<td>46.1244 dB</td>
<td>PSNR=∞ 11.3KB</td>
<td></td>
</tr>
<tr>
<td>Kodim07 10.2 KB</td>
<td>Kodim08 13.3 KB</td>
<td>Kodim15 137 KB</td>
<td>Kodim16 141 KB</td>
<td>48.6717dB</td>
<td>46.7602dB</td>
<td>PSNR=∞ 10.2KB</td>
<td></td>
</tr>
<tr>
<td>Kodim09 7.41 KB</td>
<td>Kodim10 7.41KB</td>
<td>Kodim19 146 KB</td>
<td>Kodim20 110 KB</td>
<td>48.55dB</td>
<td>41.4596dB</td>
<td>PSNR=∞ 7.41KB</td>
<td></td>
</tr>
<tr>
<td>Kodim11 10.1 KB</td>
<td>Kodim12 3.06 KB</td>
<td>Kodim23 123 KB</td>
<td>Kodim24 161 KB</td>
<td>44.7324 dB</td>
<td>46.3761 dB</td>
<td>PSNR=∞ 10.1KB</td>
<td></td>
</tr>
<tr>
<td>Kodim13 14.1 KB</td>
<td>Kodim14 11.2 KB</td>
<td>Kodim07 134 KB</td>
<td>Kodim08 176 KB</td>
<td>42.7317dB</td>
<td>47.4847dB</td>
<td>PSNR=∞ 14.1KB</td>
<td></td>
</tr>
<tr>
<td>Kodim15 3.90 KB</td>
<td>Kodim16 5.171 KB</td>
<td>Kodim11 152 KB</td>
<td>Kodim12 133 KB</td>
<td>46.8748dB</td>
<td>45.8161dB</td>
<td>PSNR=∞ 3.90KB</td>
<td></td>
</tr>
<tr>
<td>Kodim17 7.02 KB</td>
<td>Kodim18 6.91 KB</td>
<td>Kodim15 137 KB</td>
<td>Kodim16 141 KB</td>
<td>43.253dB</td>
<td>46.175dB</td>
<td>PSNR=∞ 7.02KB</td>
<td></td>
</tr>
<tr>
<td>Kodim19 6.66 KB</td>
<td>Kodim20 3.67 KB</td>
<td>Kodim23 123 KB</td>
<td>Kodim24 161 KB</td>
<td>45.1812dB</td>
<td>46.7923dB</td>
<td>PSNR=∞ 6.66KB</td>
<td></td>
</tr>
<tr>
<td>Kodim21 9.99 KB</td>
<td>Kodim22 8.18 KB</td>
<td>Kodim19 146 KB</td>
<td>Kodim20 110 KB</td>
<td>48.648dB</td>
<td>49.3497dB</td>
<td>PSNR=∞ 9.99KB</td>
<td></td>
</tr>
<tr>
<td>Kodim23 5.52 KB</td>
<td>Kodim24 11.4 KB</td>
<td>Kodim03 129 KB</td>
<td>Kodim04 143 KB</td>
<td>41.6513dB</td>
<td>47.2861dB</td>
<td>PSNR=∞ 5.52KB</td>
<td></td>
</tr>
</tbody>
</table>

Images in database are resized to 512 × 512 size. Secret and Verification images are binary images and Cover image 1 and Cover image 2 are grayscale images.
References