Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 517 – 523

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Robust Cheat-Prevention for Random Grids based Visual Secret Sharing Scheme

Sonu K. Mishra[a,*], Kumar Biswaranjan[a]

[a]*Dept of EEE, IIT Guwahati, Guwahati 781039, India*

Abstract

In conventional visual secret sharing (VSS), a secret image is encrypted into noise-like random-looking shares, a subset of which when stacked together reveal the secret. Random grids (RG) eliminate pixel expansion and extensive codebook designs of conventional VSS. However, like conventional VSS, RG-based VSS are prone to collusion attacks. Existing cheat-prevention algorithms reduce the probability of collusion attacks, but these do not consider issues arising due to noise-like nature of the shares. We introduce two issues that may lead to false accusations and we devise a robust cheat-prevention algorithm to tackle these. Experimental results validate the efficacy of the algorithm.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Visual Secret Sharing; Random Grids; Collusion

1. Introduction

Rapid advancement of technology accelerated the process of creation of digital data and solved the problem of storage. Proportionately, the preservation of digital data from being plagiarised is the need of the hour. The existing techniques in digital data preservation like watermarking, steganography, cover channel, etc. suffer from intensive computational cost during decryption. On the other hand, in visual cryptography (VC) proposed by Naor and Shamir[1] in 1995, the decryption can be done easily with human visual system without any aid of computers. A secret when distributed among numerous participants is more secure than when kept by one person – this formed

* Corresponding author. Tel.: +91-805-000-1850
E-mail address: mishrasonu1993@gmail.com

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.077

the base of VC. In (k, n)-VC, a secret image is encrypted into n share images, which individually appear to be completely random, hence carrying no information. For decryption, these images are printed on transparencies and if k, a pre-decided threshold, such shares are stacked together, then the secret image is revealed which can be easily identified with naked eyes. Various VC algorithms for secret sharing have been developed in recent years. Most of these suffer from three major drawbacks:

- *Pixel Expansion*: Each pixel in the secret image is mapped into m pixels in share images. This leads to large-sized share images making their handling and storage challenging.
- *Contrast of the reconstructed image*: The image obtained after stacking the share images is poor in contrast which makes the decryption difficult.
- *Collusion*: A group of cheaters can collude among themselves to create fake shares in order to show a false secret and hence can deceive other innocent secret holders.

The first two drawbacks can be resolved by random grid[2] (RG)-based techniques. However, like conventional VSS, the RG-based secret sharing scheme also suffers the wrath of the third problem. Most of the previous works focused on pixel expansion and contrast problems, but the collusion problem although equally important got little attention.

Existing cheat-prevention algorithms are based on authenticating the shares of each participant before using these for decrypting the secret[7]. In such schemes, n verification shares which are generated from n verification images are also distributed among the participants along with their RG share images. To verify the validity of an RG share, a participant stacks the RG share onto her own verification share. Only a genuine share when stacked with her verification share will reveal her verification image. Thus, deceitful participants can be identified and collusion can be prevented. However, these schemes do not address the problems resulting due to errors in distribution of the RG and the verification shares or stealing and deliberate infringement of shares of any particular participant. This paper presents a collusion prevention algorithm robust to these issues.

The rest of the paper is organized as follows. Section 2 describes the background and related work. Section 3 presents new problems in existing cheat-prevention schemes and proposes an algorithm to address these. The performance of the proposed algorithm is studied and evaluated against an existing cheat-prevention algorithm through extensive experiments in section 4. Section 5 concludes the paper.

2. Related Work

An RG is a 2-dimensional array of uncorrelated black/white (opaque/transparent) pixels obtained with a fair coin-toss experiment. Owing to the uniform random distribution of black and white pixels, the average light transmission of an RG is equal to ½ and hence the contrast is 0. Kafri and Keren[2] proposed an algorithm for (2, 2)-RG for binary images, which we describe in Algorithm 1. Shyu[3] extended this for grayscale and color images.

Algorithm 1: (2, 2) Random Grid[2]

Input: A binary image B
Output: Two random grid shares R_1 and R_2
for each pixel (x, y) in R_1
  $R_1(x, y) = 0/1$ randomly
end
for each pixel (x, y) in B
  if $(B(x, y) = 0), R_2(x, y) = R_1(x, y)$
  else, $R_2(x, y) = \overline{R_1(x, y)}$
end
output R_1, R_2

Algorithm 2: (n, n) Random Grid[4]

Input : A binary image B
Output : n random grids R_i, $i \in \{1,2,\ldots,n\}$
for $k \in \{1,2,\ldots,n-1\}$
  Generate R_k using coin-toss for each pixel
end
$A_1 = R_1$
for $k \in \{2,\ldots,n-1\}$
  $A_k = R_k \oplus A_{k-1}$
end
$R_n = B \oplus A_{n-1}$; *output* R_i, $i \in \{1,2,\ldots,n\}$

Shyu [4] extended the (2, 2)-RG to (*n*, *n*)-RG for binary images and also proposed extensions for grayscale and color images. The scheme for binary images is given in Algorithm 2. Chen et al.[5] proposed (*n*, *n*)-RG using a simple extension of (2, 2)-RG. First, the secret image *B* is encrypted into two share images R_1 and S_2; S_2 is encrypted into R_2 and S_3, and so on until *n* share images, $R_1 \ldots R_{n-1}$ and S_n, are generated. The scheme is shown in Fig. 1. Chen et al.[6] extended (*n*, *n*)-RG to (*k*, *n*)-RG. First, *k* interim random shares are generated using a (*k, k*)-RG. And then, pixels in these shares are distributed randomly among *n* other random shares. The scheme is described in Algorithm 3.

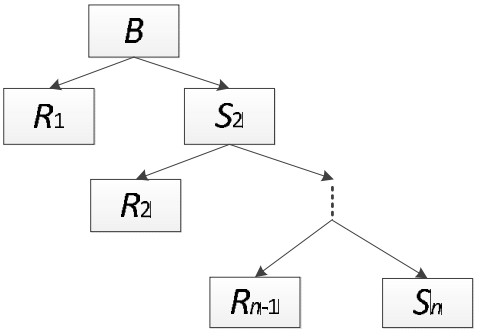

Fig. 1. Flow chart describing (*n, n*)-RG[5].

Algorithm 3: (*k*, *n*) Random Grid[6]

Input : A binary image *B*

Output : *n* Random Grids R_i, $i \in \{1,2,\ldots,n\}$

Generate *k* Random Grids I_i, $i \in \{1,2,\ldots,k\}$ using (k,k) RG

Generate *n* Random Grids R_i, $i \in \{1,2,\ldots,n\}$ using coin toss

for each pixel (x, y) in *B*

vector = select *k* distinct integers in $\{1, 2,\ldots,n\}$

$R_{\text{vector}}(x,y) = I(x,y)$

end

output R_i, $i \in \{1,2,\ldots,n\}$

Horng et al.[7] tried to address the collusion problem by: (1) employing a cheat-prevention scheme which validates the veracity of the shares using verification shares, (2) distributing *n* out of (*n*+*l*) shares of the (2, *n*+*l*) threshold VSS among the participants to reduce the probability of leaking the structure of the shares. In the genetic algorithm (GA)-based VSS scheme proposed by Tsai et al.[8], several homogeneous secret images are used to deter the complicity. The secret images are such that all sets of two shares divulge different secret images. In a generic approach proposed by Hu and Tzeng[9], a VSS scheme is modified to another VSS scheme possessing cheat-prevention ability. Chang et al.[10] proposed a cheat-prevention scheme with embedding of an authentication image in each share and recovering it from shares for their authentication. All the above schemes were proposed for conventional VSS. Lee and Chen[11] proposed collusion attacks which will work on (2, *n*) and (*k*, *n*)-RG based VSS. Wu and Sun[12]have presented a cheat-prevention scheme to counterattack the collusion problems of Lee and Chen[11].

In this paper, we introduce new problems arising due to the RG and the verification shares being completely random; these are not considered in the schemes proposed by Horng et al.[7] and Wu and Sun[12]. We discuss a possible algorithm to alleviate such problems and then we prove the effectiveness of our algorithm by experimentation.

3. New Problems in Cheat-Prevention

We consider a threshold (*k, n*)-RG based VSS, i.e. a secret is broken into *n* RG shares and given to a group of *n* participants; whenever at least *k* of them would come together and stack their shares, they would get to see the secret. Now, if the group has *k* cheaters, $c_1 \ldots c_k$, then they can stack their shares, see the actual secret and then can modify their shares to encrypt a false secret. When these *k* cheaters approach an innocent participant *I* with their modified shares, on stacking their shares with *i*'s share, they would show the false secret to *i*. Thus, the participant *I* would be deceived.

Therefore, a participant should verify the veracity of the shares before stacking these in her attempt to see the secret. Horng et. al[7] proposed the use of verification shares for this integrity check. First, *n* RG shares, R_i, are generated and *n* verification images, V_i, are obtained for $i \in \{1 \ldots n\}$. Using these RG shares and verification images, *n* verification shares, S_i, are generated and each participant *i* is given a set, $\{R_i, V_i, S_i\}$ where $i \in \{1 \ldots n\}$, of shares. These verification shares and verification images can be used to check the authenticity of the shares. This idea was initially proposed for conventional VC-based secret sharing, but was later used by Wu and Sun[12] to counterattack the collusion problem in RG-based VSS proposed by Lee and Chen[11].

Wu and Sun's[12] method for cheat-prevention in RGs generate S_i as $S_i = f(V_i, R_j; j \in \{1 \ldots n\} - \{i\})$. This method suffers from following problems:

- *Problem* P_1: If due to randomness in appearance of all the shares, a participant (say j) gets a wrong verification share (say S_k) then the whole verification procedure will become intractable. Even if an honest participant i brings her original RG share R_i, on stacking with S_k, it will not yield V_j. Then i will be declared a cheater.
- *Problem* P_2: If someone from inside or outside of the group gets access to the i's shares and replaces them with some other random shares, the current scheme offers no method to i to verify the authenticity of her own shares. Since S_i and R_i anyway do not produce V_i, even if R_i is replaced with other share, nothing can be verified.

Problem P_1 can be circumvented by using extended visual cryptography scheme[13,14] (EVCS) where distinct cover images are stamped onto the random looking RG shares, but this negatively affects the visual quality of the reconstructed images. Also, EVCS cannot solve problem P_2, because a different RG share stamped with the same cover image can be used to cheat. Therefore, we propose to make S_i a function of V_i and all share images, i.e. $S_i = f(V_i, R_j; j \in \{1 \ldots n\})$. Using this modification, S_i and R_i can be used to see if they produce V_i. If they do not, then participant i can ask the dealer to redistribute the shares. Our algorithm is described in pseudo-code of Algorithm 4.

Algorithm 4: Robust Cheat-prevention Algorithm for RGs

Input : Secret Shares R_i, Verification Images V_i, $i \in \{1,2,\ldots,n\}$
Output : Verification Shares S_i, $i \in \{1,2,\ldots,n\}$
for $i = 1$ *to* n
  for each pixel (x, y) in V_i
    $u \leftarrow$ randomly chosen from $\{1,2,\ldots,n\}$
    if $V_i(x, y)$ is white
      $S_i(x, y) = R_u(x, y)$
    else
      $S_i(x, y) = \overline{R_u(x, y)}$
    end
  end
end
output S_i, $i \in \{1,2,\ldots,n\}$

Our contribution is twofold: First, we introduce new problems—the risk of incorrect distribution of shares among the participants and the risk of deliberate altercation to the shares of a participant—that may lead to wrong accusations in existing cheat-prevention algorithms and we devise an algorithm to prevent these. Second, we improve the contrast of the reconstructed images as compared to Wu and Sun[12]. The improvement in contrast of reconstructed images in Algorithm 4 comes from the fact that when a pixel in V_i is white, the corresponding pixel in S_i is assigned the same value as in R_u, unlike the algorithm by Wu and Sun[12] where it is assigned randomly.

4. Experiments and Results

The performance of the proposed algorithm is studied experimentally for binary images and evaluated against the algorithm of Wu and Sun[12]. We used a threshold (k, n)-RG based VSS where the number of participants is $n = 4$ and the minimum number of shares required to reconstruct the secret is $k = 3$. Genuine secret, false secret and verification images $\{V_1, V_2, V_3, V_4\}$ considered for the experiments are shown in Fig. 2. The resultant RG shares $\{R_1, R_2, R_3, R_4\}$ and the verification shares $\{S_1, S_2, S_3, S_4\}$ generated using Algorithm 4 are given in Fig. 3. It can be observed that all the RGs and the verification shares are completely random having no trace of the secret image.

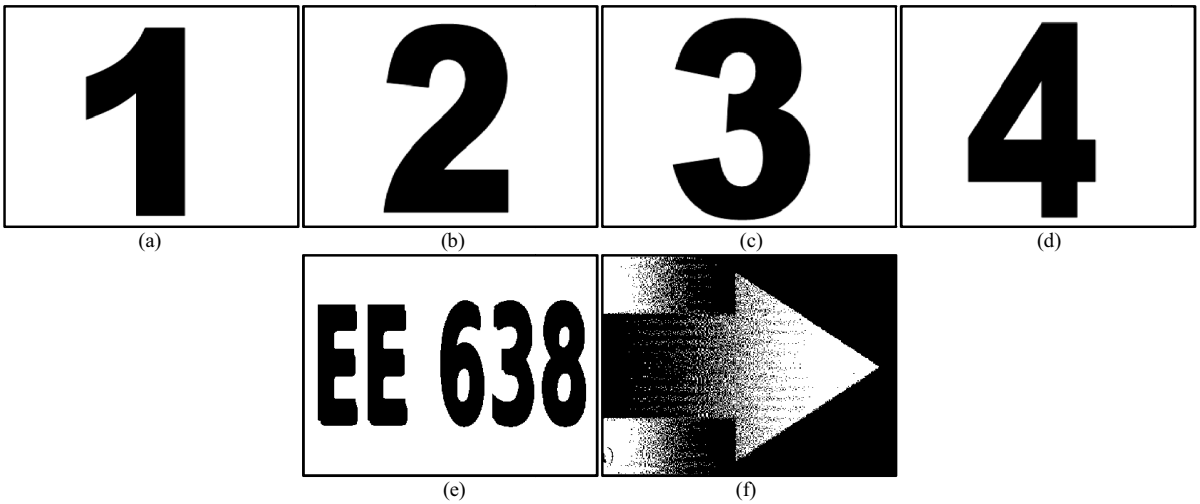

Fig. 2. (a) verification image V_1; (b) verification image V_2; (c) verification image V_3; (d) verification image V_4; (c) genuine secret image; (d) false secret image.

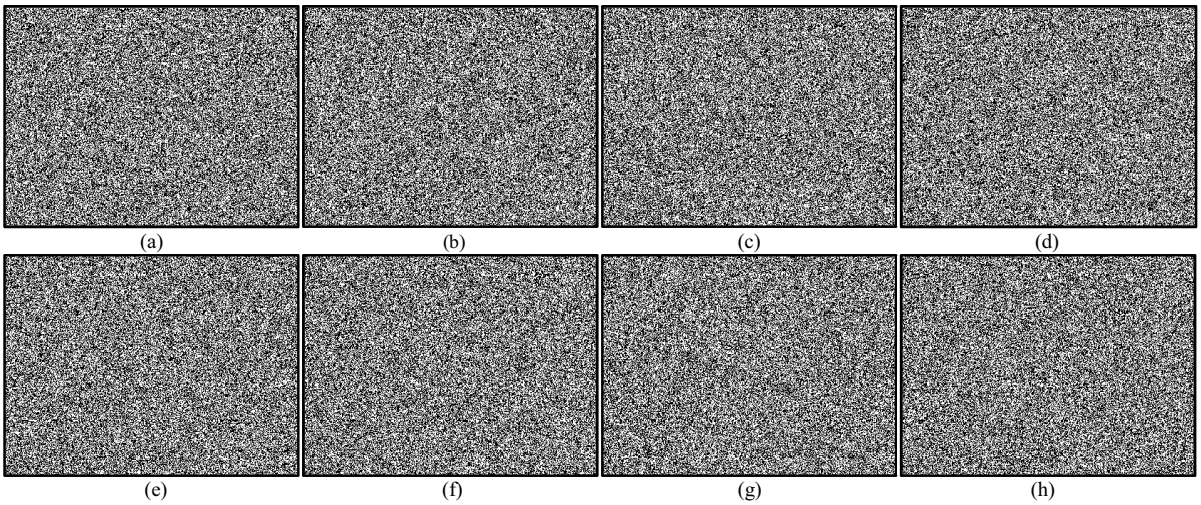

Fig. 3. (a) RG share R_1; (b) RG share R_2; (c) RG share R_3; (d) RG share R_4; (e) verification share S_1; (f) verification share S_2; (g) verification share S_3; (h) verification share S_4.

We assume that participants 2, 3, 4 are cheaters who want to deceive participant 1 by showing her the false secret. They stack their RG shares, see the actual secret and then modify their RG shares R_2 and R_3 to fake shares F_2 and F_3 in order to encrypt the false secret. The fake shares and the results of various stacking combinations are shown in Fig. 4. The fake shares being completely random are difficult to be differentiated from the real shares. Fig. 4(d) shows that the participants 2 and 3 have modified their shares such that these show the false secret when stacked with R_1, the share of innocent participant 1. However, the participant 1, acquainted with this possibility, will first stack the shares F_2 and F_3 onto the verification share S_1. It is evident that these fake shares do not yield the verification image V_1 when stacked with S_1 (Fig. 4. e, f). Hence, the participant 1 detects the collusion and saves her from being cheated by not involving in any further secret sharing with the other two. On the other hand, if the participants 2 and 3 are honest and bring their original RG shares R_2 and R_3, the stacks of R_2 and R_3 with S_1 yield the verification image V_1, authenticating the shares (Fig. 5. a, b). Then, the participant 1 stacks her RG share R_1 with those of other two R_2 and R_3in order to see the real secret (Fig. 5. c). Thus, our cheat prevention algorithm is capable of locating deceitful participants and hence is robust to collusion attacks.

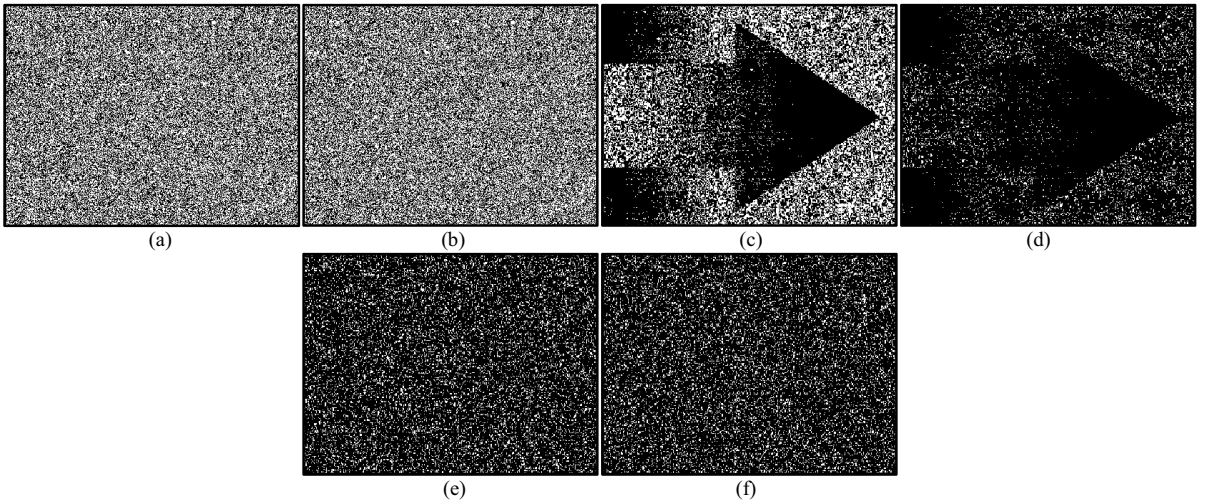

Fig. 4. (a) fake share F_2; (b) fake share F_3; (c) stack $F_2 + F_3$; (d) stack $F_2 + F_3 + R_1$; (e) stack $S_1 + F_2$; (f) stack $S_1 + F_3$.

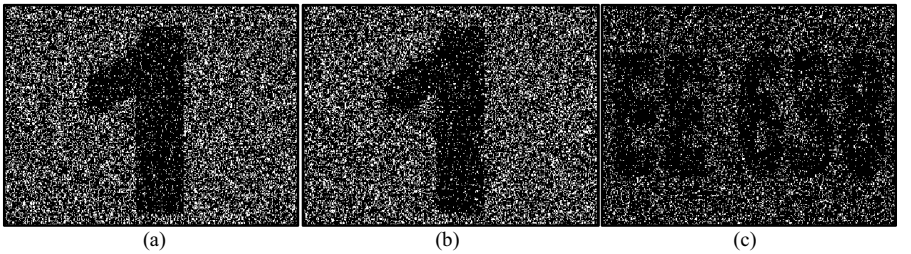

Fig. 5. (a) stack $S_1 + R_2$; (b) stack $S_1 + R_3$; (c) stack $R_1 + R_2 + R_3$.

Here the performance of proposed algorithm is compared with that of the algorithm of Wu and Sun[12]. Algorithm by Wu and Sun[12] has following problems.

- *Poor Contrast*: In Fig. 6 (a, b), the results of S_1+R_2 stacks are shown for Algorithm 4 and Wu and Sun[12]. It can be seen that the reconstructed verification image produced by Algorithm 4 has much better contrast as compared to that produced by Wu and Sun[12].
- *Distribution Errors*: Let us assume that the dealer mistakenly gave participant 1 V_1, R_1 and R_2 (in place of S_1). If the verification shares $\{S_1, S_2, S_3, S_4\}$ are obtained using Wu and Sun[12] algorithm, the stacks of R_1 and R_2, and R_1 and S_1 both give completely random looking results (Fig 6. c, d). Hence, the participant 1 cannot detect the mistake in distribution. However, if Algorithm 4 is used for obtaining the verification shares, the noise-like random result corresponding to stack R_1 and R_2 (Fig. 6. e) will alarm the participant 1 because R_1 and S_1 stack should show V_1 (Fig. 6. f). Hence the distribution error can be easily detected. The participant 1 can then notify this to the dealer and ask for re-distribution of the shares.
- *Stealing and Forging*: Now let us assume that someone gets access to the participant 1's RG share R_1 and replaces that with a completely random share P_1. The algorithm by Wu and Sun[12] cannot detect this breach because the stack of R_1 and S_1, and P_1 and S_1 are similar – completely random (Fig. 6. d, g). However, Algorithm 4 can easily differentiate between the two, because the stack of P_1 and S_1 yield a noise-like random image unlike the stack of R_1 and S_1 which shows V_1. The results obtained from Algorithm 4 are shown in (Fig. 6. f, h). Hence, this security breach can be easily discovered. The other participants and the dealer can be notified; the dealer can further decide to generate new RG and verification shares.

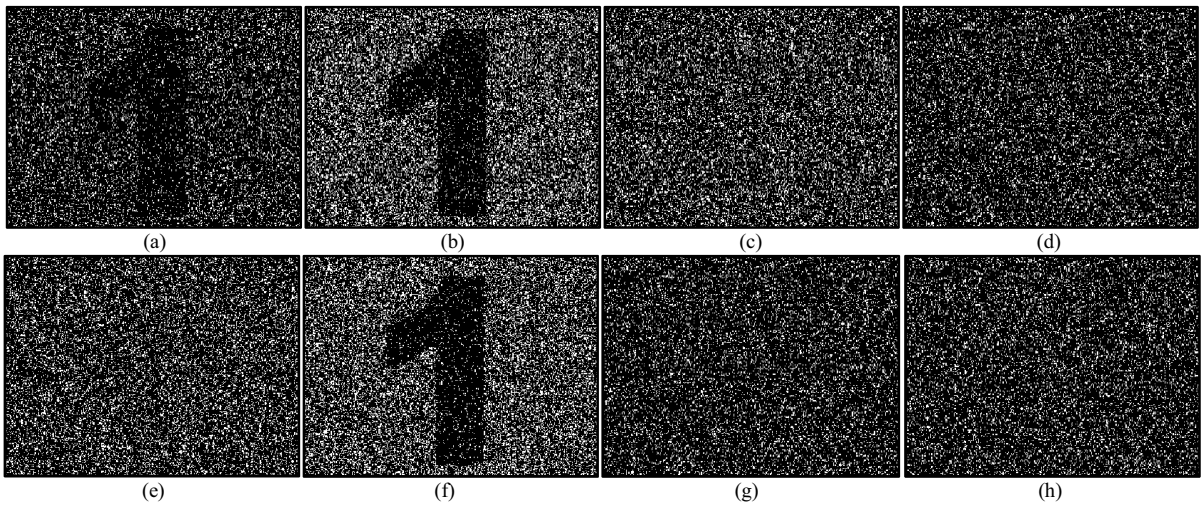

Fig. 6. (a) $S_1 + R_2$, Wu and Sun[12]; (b) $S_1 + R_2$, Algorithm 4; (c) $R_1 + R_2$, Wu and Sun[12]; (d) $R_1 + S_1$, Wu and Sun[12]; (e) $R_1 + R_2$, Algorithm 4; (f) $R_1 + S_1$, Algorithm 4; (g) $S_1 + P_1$, Wu and Sun[12]; (h) $S_1 + P_1$, Algorithm 4.

5. Conclusion

In this paper, we presented management problems—the risk of incorrect distribution of shares among the participants and the risk of deliberate altercation to the shares of a participant—caused by the randomness of the shares involved leading to wrong accusations in the existing cheat-prevention algorithms. The importance of the problems and how they can affect secret sharing schemes were shown. An algorithm to address these issues was proposed which was shown to be effective with help of meticulous experimentation. The algorithm also produced better contrast reconstructed images as compared to those obtained from existing algorithm.

References

1. Naor M, Shamir A. Visual cryptography. *Eurocrypt*; 1995; **950:**p. 1-12.
2. Kafri O, Keren E. Encryption of pictures and shapes by random grids. *Optics Letters*; 1987; **12:**6:p. 377-379.
3. Shyu SJ. Image encryption by random grids. *Pattern Recognition*; 2007; **40:**p. 1014-1031.
4. Shyu SJ. Image encryption by multiple random grids. *Pattern Recognition*; 2009; **42:**p. 1582-1596.
5. Chen TH, Tsao KH. Visual secret sharing by random grids revisited. *Pattern Recognition*; 2009; **42:**p. 2203-2217.
6. Chen T, Tsao, K. Threshold visual secret sharing by random grids. *Journal of Systems and Software*; 2011; **84:**p. 1197-1208.
7. Horng G, Chen T, Tsai D. Cheating in visual cryptography. *Designs, Codes and Cryptography*; 2006; **38:**p. 219-236.
8. Tsai D, Chen T, Horng GA cheating prevention scheme for binary visual cryptography with homogeneous secret images. *Pattern Recognition*; 2007; **40:**p. 2356-2366.
9. Hu C, Tzeng W. Cheating prevention in visual cryptography. *IEEE Transactions on Image Processing*; 2007; **16:**p. 36-45.
10. Chang C, Chen T, Liu L. Preventing cheating in computational visual cryptography. *Fundamenta Informaticae*; 2009; **92:**p. 27-42.
11. Lee Y, Chen T. Insight into collusion attacks in random-grid based visual secret sharing. *Signal Processing*; 2012; **92:**p. 727-736.
12. Wu X, Sun W. Random grid-based visual secret sharing for general access structures with cheat-preventing ability. *Journal of Systems and Software*; 2012; **85:**p. 1119-1134.
13. Lee KH, Chiu PL. An Extended Visual Cryptography Algorithm for General Access Structures. *IEEE Transactions on Information Forensics and Security*; 2012; **7:**1:p. 209-219.
14. Guo T, Liu F, Wu CK. k out of k extended visual cryptography scheme by random grids. *Signal Processing*; 2014; **94:**p. 90-101.