The Journal of Supercomputing (2024) 80:23205–23224
https://doi.org/10.1007/s11227-024-06334-z

Exploring recent advances in random grid visual cryptography algorithms

Neetha Francis[1] · A. Lisha[1] · Thomas Monoth[1]

Accepted: 29 June 2024 / Published online: 10 July 2024

Abstract
Visual cryptography scheme is initiated to securely encode a secret image into multiple shares. The secret can be reconstructed by overlaying the shares, making it perceptible to the human visual system. Random grid visual cryptography offers key advantages over traditional visual cryptography such as avoiding pixel expansion and the requirement for intricate codebook design. This paper is intended to conduct a critical analysis and implementation of various algorithms in random grid visual cryptography from 2011 to 2022. The focal parameters of this study are theoretical contrast, experimental contrast and visual quality. Experiments were done on binary images for *(k, n)* algorithms considering different values of threshold. By implementing and analysing the results, the scheme which gives the optimal result is identified.

Keywords Visual cryptography · Random grid · Contrast · Pixel expansion · Codebook design · Aspect ratio

1 Introduction

Visual cryptography is an innovative domain of cryptography that combines the power of encryption with the visual representation of information. Ever since the technology confronted the need to transfer digital data, securing it with reliable techniques were pursued. Traditional cryptographic methods were evolved so as to

✉ Neetha Francis
neethapradeep@gmail.com

A. Lisha
lisha@marymathacollege.ac.in

Thomas Monoth
tmonoth@yahoo.com

1 Department of Computer Science, Mary Matha Arts & Science College, Mananthavady, Wayanad, Kerala, India

manipulate on texts or numerical data. Further inventions were made using the visual elements like images and patterns. This niche domain of cryptography in which encryption is done using the visual representation of information is termed as Visual Cryptography Scheme (VCS). The techniques in visual cryptography are widely used today where privacy and confidentiality are of great significance. Visual cryptography has practical applications in secure communications, document authentication and watermarking allowing sensitive information to be shared and verified through multiple image shares. However, it faces challenges such as pixel expansion, ensuring high quality reconstructed images and managing color image processing. The fundamental idea of VCS was explained by Naor and Shamir [1] in 1994. Since then, it invited a great attention of many researchers from different domains such as image authentication, watermarking and information security. The overview of visual cryptography is to split an image into parts called shares and any individual share does not suffice to retrieve the secret image [2]. The individual shares are then distributed to different shareholders. The secret image can be recouped when the shares are combined together by stacking.

Visual cryptography is a method in which a secret image is shared within a group of shareholders. The image which is the secret is entrusted with group members as unusable fragments. Unlike conventional cryptographic techniques that rely on complex algorithms and secret keys, visual cryptography does not require any cryptographic expertise to retrieve the original image. This makes it applicable for scenarios where users may have limited technical knowledge.

A (k, n) VCS involves splitting up of a secret image into n parts. The reconstruction of the secret image becomes feasible when a minimum of k segments are combined. However, if less than k segments are involved, the reconstructed image will be meaningless and insignificant. Decoding is achieved by the Human Visual System (HVS) without requiring intricate computations. This feature is widely regarded as the foremost advantage of VCS [3]. The conventional VCS have the following drawbacks: (i) pixel expansion (ii) complex codebook design (iii) change in aspect ratio [4].

Kafri and Keren [5] pioneered the encryption of binary images by Random Grid Visual Cryptography Scheme (RGVCS) in 1987. A binary image is split into two random parts such that they have same dimension as that of the binary image. RGVCS can also be used for generating master share and private share for each participant to enhance the security and confidentiality [6]. It exceeds conventional VCS by preserving the secret image's original size, effectively eliminating the issue of pixel expansion. This paper reviewed and analysed existing *(k, n)* random grid algorithms with different values of threshold.

2 Random grid visual cryptography scheme

This section gives some definitions and a concise overview of traditional RGVCS. Consider $\oplus$ as OR and $\otimes$ as XOR. In order to encrypt a binary secret image, S of dimension $M \times N$ with $S(i, j)$, $1 \leq i \leq M$, $1 \leq j \leq N$, is allotted to n shares $S_1, S_2, S_3, \ldots, S_n$ and from t $(1 \leq t \leq n)$ shares the secret image S' is retrieved. Kafri and Keren [5] described random grid as a 2-D array of pixels that is formed as whether the pixels are transparent or not. There is an equal chance of having transparent or opaque pixels as the decision is made using a coin-flip method [5]. For example, consider a pixel b within a randomly generated grid R. The likelihood of b being transparent is the same as the likelihood of it being opaque. $Prob\ (b=0)=1/2=Prob\ (b=1)$, where $b=0$ indicates transparent and $b=1$ means opaque. The light transmission of b is denoted as $\pounds(b)=1/2$.

Consider S_1 and S_2, two standalone random grids with same size, $s_1 \in S_1$ and $s_2 \in S_2$ where s_1 and s_2 are in same position of S_1 and S_2. When these two random pixels s_1 and s_2 are superimposed with OR, only four combinations are possible. For $s_1 \oplus s_2$ being transparent, the probability is $1/4$. Since S_1 and S_2 are superimposed, the average light transmission of S_1 and S_2 (s_1 and s_2) is $1/4$. $\pounds(\ s_1 \oplus s_2)=1/4$ and ($\pounds(S_1 \oplus S_2)=1/4)$.When $\overline{b}$ is a complement of b, $b \oplus \overline{b}=1 \Rightarrow Prob\ (b \oplus \overline{b}=\ 0)=0.$ Therefore $\pounds(b \oplus \overline{b})=0$ or $\pounds(S \oplus \overline{S})=0.$

Definition 1 (*Average light transmission*): The light transmission of a transparent (resp. opaque) pixel is expressed as $\pounds(b)=1$ (resp. 0) for a specific pixel b in a binary image S of size M x N. Overall, the general average light transmission of S is described as [7,8].

$$\pounds(S) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \pounds(S(i,j))}{MXN} \tag{1}$$

Definition 2 (*Contrast*): The contrast (α) between the original image S and the recovered binary image S' is defined as [7,8]

$$\alpha = \frac{\pounds(S'[S(0)] - \pounds(S'[S(1)]}{1 + \pounds(S'[S(1)]} \tag{2}$$

Three algorithms were introduced by Kafri & Keren [5] to generate two meaningless random shares S_1 and S_2. *Random_GridAlg1* is considered to be better while contrast and light transmission are considered[7]. In encoding, S_1 is created by choosing the values from normal distribution. Concurrently, the value of a specific pixel in S_2 is decided by the value of respective pixel in the secret image. If the value is transparent, then S_1 and S_2 have the same value; or else they have complementary values. The resultant grid $S_1 \oplus S_2$ always yields 1 when the pixel of S is black, and it has a partial probability of being transparent or opaque when the pixel of S is white. Kafri and Keren's [5] algorithms are explained in detail.

Input **:** Secret image S

Output*:* Random grids S_1 & S_2

$$\text{where } S_1 = \{S_1(x, y) \mid S_1(x, y) \in \{0, 1\}, 1 \le x \le M, 1 \le y \le N$$

$$S_2 = \{S_2(x, y) \mid S_2(x, y) \in \{0, 1\}, 1 \le x \le M, 1 \le y \le N$$

Random_GridAlg1

Read secret image S

Generate a random grid S_1 *where* $S_1(x, y) \in \{0, 1\}$

For every pixel in S,

$$\text{Compute } RG_2(x, y) = S_1(x, y), \text{if } S(x, y) = 0$$
$$= \sim S_1(x, y), \text{if } S(x, y) = 1$$

Output S_1 & S_2

Random_GridAlg2

Read secret image S

Generate a random grid S_1 *where* $S_1(x, y) \in \{0, 1\}$

For every pixel in S,

$$\text{Compute } RG_2(x, y) = S_1(x, y) \text{ if } S(x, y) = 0$$
$$\in \{0, 1\}, \text{ if } S(x, y) = 1$$

Output S_1 & S_2

Random_GridAlg3

Read secret image S

Generate a random grid S_1 *where* $S_1(x, y) \in \{0, 1\}$

For every pixel in S,

$$\text{Compute } S_2(x, y) \in \{0, 1\}, \text{ if } S(x, y) = 0$$
$$= \sim S_1(x, y), \text{ if } S(x, y) = 1$$

Output S_1 & S_2

The recovered images using the above algorithms are shown in Fig. 1. The equations for finding Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) and Structured Similarity Index Measure (SSIM) are given.

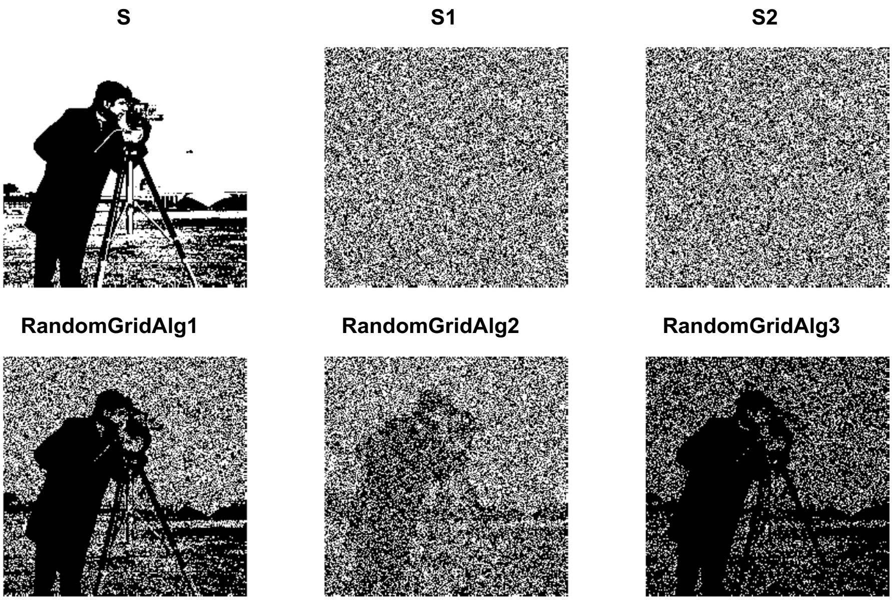


Fig. 1 Reconstructed images proposed by Kafri & Keren [5]

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j) - K(i,j)\right]^2 \tag{3}$$

where m and n represents the dimensions of the images, I denotes the original image, and K signifies the distorted image.

$$\mathrm{PSNR} = 10 - \log 10\left(\frac{MAX^2}{MSE}\right) \tag{4}$$

where MAX represents the highest possible pixel value in the image.

$$\mathrm{SSIM(a,b)}\frac{\left(2\mu_a\mu_b + C_1\right)\left(2\sigma_{ab} + C_2\right)}{\left(\mu_a^2 + \mu_b^2 + C_1\right)\left(\sigma_a^2 + \sigma_b^2 + C_2\right)} \tag{5}$$

where μ_a is the mean of a, μ_b is the mean of b.

$\sigma^2{}_a$ and $\sigma^2{}_b$ are the variances of a and b.

σ_{ab} is the covariance between a and b.

$C_1=(k_1L)^2$ and $C_2=(k_2L)^2$ are constants. PSNR and SSIM values for the above algorithms are given in Table 1.

Despite the superiority of RGVCS over traditional VCS, the former received less attention until Shyu extended the schemes [3,8]. Chen and Tsao [7,10] presented *(n, n)* and *(2, n)* schemes for binary and color images.

Table 1 PSNR & SSIM

	Random_GridAlg1	Random_GridAlg2	Random_GridAlg3
PSNR	53.3951	51.6517	52.1514
SSIM	0.5142	0.4183	0.4695

Random_Grid (S, n) demonstrates the *(n, n)* scheme in which *Random_Grid()* can be substituted by *Random_GridAlg1*, *Random_GridAlg2* or *Random_GridAlg3*. *Rand_Perm()* randomly shuffles and assigns the generated bits to the output shares.

(n, n) RGVCS-Random_Grid (S, n)

Create SG_1, S_2 as two random grids

$SG_1||S_2$ = *Random_GridAlg1(S)*

Create $SG_2 \sim SG_n$

if (n >2)

for k = 2 to n-1

$SG_k||S_{k+1}$ = *Random_GridAlg1*(S_k)

Create SG_n as the last random grid

$SG_n = S_n$

3 Recent research advances in *(k, n)* random grid schemes

Chen and Tsao [9] extended the basic VCS techniques of *(2, 2)* by Kafri and Keren [5], *(2, n)* and *(n, n)* by Chen & Tsao [7,10], Shyu [3,8] to *(k, n)* VCS. The shareholders should provide no less than *k* individual shares to get back the secret. Fewer than *k* shares produce meaningless superimposed image. Calculations are also done to determine the theoretical contrast of the superimposed result. The feasibility of the suggested method is demonstrated by simulating *(2, 4)* and *(3, 4)* schemes for both binary and color images.

Wu and Sun [11] extended Chen and Tsao's method [9] to obtain a contrast-improved VCS on random grids and presented a void and cluster based post-processing to increase the uniformity of the recovered image. The visual quality of the retrieved image is increased by using the recommended schemes. Optimal contrast is attained by the contrast- improved RGVCS and an improved image can be recovered by using the post-processing techniques.

Lee et al. [12] presented quality-improved random grid based VCS scheme which outperforms Chen & Tsao's (*k, n*) algorithm [9] as its visual quality is a concern. Rather than arbitrarily generating *n-k* pixels, they are selected from the generated *k* pixel values. Further investigations and findings of contrast proved that the proposed algorithm surpasses the existing techniques.

A unique *(k, n)* RG-based VCS proposed by Guo et al. [13] enhances the contrast of Chen and Tsao's algorithm [9], thereby increases the visual quality. However, the values of *k* and *n* do not accurately describe the scheme's contrast. The experiments and results showed the superiority of this approach over Chen and Tsao's scheme.

Liu et al. [14] presented a (*k, n*) threshold VCS based on random grids, which has improved visual quality when compared to Wu and Sun's [11] and Guo et al.'s [13] algorithms. This paper analysed the features that determine the visual quality and utilized the generated bits to improve the parameters. The feature of attaining improved contrast causes this algorithm to be applied for other domains also.

Yan et al. [15] introduced a novel VCS threshold scheme in which multiple decryptions based on OR and XOR are considered. The original image can be fully recuperated if there is a machine having XOR operation is available. The image can also be retrieved with good quality by HVS without any complex cryptographic algorithms when minimum number of shares is available for *(k, n)* VCS scheme. Several experiments were performed to assess the effectiveness of the recommended scheme.

Hu et al. [16] introduced a contrast enhanced scheme for *(k, n)* VCS based on random grids and a common equation for calculating theoretical contrast with better accuracy. It was inferred that there is some relationship between the first *k* pixels and the contrast. As a result, using the initial *k* pixels, it is possible to produce the final *n-k* pixels. Theoretical study and findings from the experiments indicates that the suggested method has improved visual quality and higher value of contrast than the existing threshold techniques.

Zhao and Fu [17] developed a new contrast enhanced *(k, n)* VCS based on OR and XOR decryptions. A common formula to calculate the value of theoretical contrast is also mentioned in the paper. The recommended scheme is using the parity basis matrices and hence there is no pixel expansion. When suitable computing devices are available, an XOR procedure can completely recoup the original image. Further investigations and studies guaranteed the efficiency and security of the method.

4 Implementation and analysis of (*k, n*) random grid schemes

We have implemented different *(k, n)* VCS techniques based on random grids using Python. Evaluation parameters such as theoretical contrast (α), experimental contrast (τ) and visual quality are considered to validate the efficiency of each algorithm. A binary image S of size $M \times N$ is the input to Algorithms 1–8 and n random grids S_1, $S_2, \ldots, S_n$ of the same size are generated. The different schemes are two-out-of-two, two-out-of-n, n-out-of-n and k-out-of-n. This paper reviewed and analysed (k, n) algorithms of various researchers during 2011 to 2022.

Input: Secret image S.

Output: random grids $S_1, S_2, \ldots, S_n$.

Algorithm 1

Step 1.1 $\forall$ S(i, j), where $1 \le i \le M$, $1 \le j \le N$, repeat steps 1.2 and 1.3

Step 1.2 Generate $r_1, r_2, \ldots, r_k$

$$[r_1, r_2, \ldots, r_k] = Random\_Grid\,(S\,(i, j), k, n)$$

Step 1.3 Generate n-k bits

$$[r_{k+1}, r_{k+2}, \ldots, r_n] \in \{0, 1\}$$

Step 1.4 Randomly rearrange and distribute n bits

$$[S_1, S_2, \ldots, S_n] = Random\_Perm\,(r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$$

The theoretical contrast for the algorithm is given by the equation

$$\alpha = \frac{2 * tC_k}{(2^t + 1)nC_k - tC_k} \text{where } k \le t \le n \tag{6}$$

For $(3, 5)$,$t = 3 \rightarrow \alpha = \frac{2*3C_3}{(2^3+1)5C_3-3C_3} = \frac{2}{89} = 0.0225$

$$t = 4 \rightarrow \alpha = \frac{2 * 4C_3}{(2^4 + 1)5C_3 - 4C_3} = \frac{8}{116} = 0.0482$$

$$t = 5 \rightarrow \alpha = \frac{2 * 5C_3}{(2^5 + 1)5C_3 - 5C_3} = \frac{1}{16} = 0.0625$$

Algorithm 1[9] makes use of the basic algorithm to generate k bits and the rest of n-k bits are arbitrarily chosen from normal distribution. By randomly rearranging and permuting n bits, n shares are obtained. The theoretical contrast for $(2, 4)$ scheme is 0.0690, 0.1176, 0.125 for $t=2, 3, 4$ respectively and for $(3, 5)$ scheme the value is 0.0225, 0.0482, 0.0625 for $t=3, 4, 5$ respectively.

Algorithm 2

Step 2.1 $\forall\, S(i, j)$, where $1 \leq i \leq M$, $1 \leq j \leq N$, repeat steps 2.2 to 2.4

Step 2.2 Generate $r_1, r_2, \ldots, r_{k-1}$

$[r_1, r_2, \ldots, r_{k-1}] \in \{0, 1\}$

Step 2.3 Compute r_k

$r_k = S(i, j) \oplus r_1 \oplus r_2 \oplus \ldots \oplus r_{k-1}$

Step 2.4 Generate n-k bits

Set $r_{k+1} = r_k$, $r_{k+2} = r_k, \ldots, r_n = r_k$

Step 2.5 Randomly rearrange and distribute n bits

$[S_1, S_2, \ldots, S_n] = Random\_Perm\,(r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$

The theoretical contrast for the algorithm is given by the equation

$$\alpha = \begin{cases} (\frac{1}{2})^{k-1} for\, t = n \\ \dfrac{\binom{n-k+1}{t-k+1}^{(\frac{1}{2})^{k-1}}}{\binom{n}{t}^{+A.B.D}} otherwise \end{cases} \tag{7}$$

$$where\ \ A = \begin{cases} \left[\binom{k-1}{q-n+k-1}, \ldots\ldots, \binom{k-1}{k-1}\right] & for\, n-k+1 \leq q < n \\ \left[\binom{k-1}{0}, \ldots\ldots, \binom{k-1}{k-1}\right] & otherwise \end{cases}$$

$$B = \begin{cases} \left[\binom{n-k+1}{n-k+1}, \ldots\ldots, \binom{n-k+1}{q-k+1}\right] & for\, n-k+1 \leq q < n \\ \left[\binom{n-k+1}{q}, \ldots\ldots, \binom{n-k+1}{q-k+1}\right] & otherwise \end{cases}$$

$$D = \begin{cases} \left[(\frac{1}{2})^{q-n+k}, \ldots\ldots, (\frac{1}{2})^{k-2}, \quad (\frac{1}{2})^{k-1}, 0\right] & for\, n-k+1 \leq q < n \\ \left[\frac{1}{2}, \ldots\ldots, (\frac{1}{2})^{k-2}, (\frac{1}{2})^{k-1}, 0\right] & otherwise \end{cases}$$

For *(3, 5)*, $t = 3 \rightarrow \alpha = \dfrac{\binom{n-k+1}{t-k+1}(\frac{1}{2})^{k-1}}{\binom{n}{t}+A.B.D} = \dfrac{\binom{3}{1}(\frac{1}{2})^{2}}{\binom{5}{3}+A.B.D} = \frac{1}{16} = 0.0625$

$$t = 4 \rightarrow \alpha = \frac{\binom{n-k+1}{t-k+1}(\frac{1}{2})^{k-1}}{\binom{n}{t}+A.B.D} = \frac{\binom{3}{2}(\frac{1}{2})^{2}}{\binom{5}{4}+A.B.D} = \frac{3}{22} = 0.1364$$

$$t = 5 \rightarrow \alpha = (\frac{1}{2})^{k-1} = \frac{1}{4} = 0.25$$

In Algorithm 2 [11], the first *k-1* bits are arbitrarily chosen from normal distribution. r_k is generated by performing XOR with *S(i, j)* and $r_1, r_2, \ldots r_k$. The rest of *n-k* bits are computed by assigning r_k to $r_{k+1}, r_{k+2,\ldots}\ r_n$. The *n* bits are shuffled and rearranged to obtain the random grids. The theoretical contrast obtained for *(2, 4)* scheme is 0.2, 0.333, 0.5 for *t=2,3,4* and for *(3, 5)* scheme the value is 0.0625, 0.1364, 0.25 for *t=3,4,5* respectively.

Algorithm 3

Step 3.1 ∀ *S(i, j),* where $1 \le i \le M,\ 1 \le j \le N$, repeat steps 3.2 and 3.3

Step 3.2 Generate $r_1, r_2, \ldots, r_k$

$[r_1, r_2, \ldots r_k] = Random\_Grid1(S(i, j), k, n)$

Step 3.3 Generate *n-k* bits

$[r_{k+1}, r_{k+2}, \ldots, r_n]$ = Randomly select from $\{r_1, r_2, \ldots, r_k\}$

Step 3.4 Randomly rearrange and distribute *n* bits

$[S_1, S_2, \ldots, S_n] = Random\_Perm\ (r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$

The theoretical contrast for the algorithm is given by the equation

$$\alpha = \frac{1}{2^{k-1}} * \frac{t-k+1}{n-k+1} \tag{8}$$

For *(3, 5)*, $t = 3 \rightarrow \alpha = \frac{1}{2^{3-1}} * \frac{3-3+1}{5-3+1} = \frac{1}{2} = 0.0833$

$$t = 4 \rightarrow \alpha = \frac{1}{2^{3-1}} * \frac{4-3+1}{5-3+1} = \frac{1}{6} = 0.1666$$

$$t = 5 \rightarrow \alpha = \frac{1}{2^{3-1}} * \frac{5-3+1}{5-3+1} = \frac{1}{4} = 0.25$$

Algorithm 3 [12] generates *k* bits by *Random_Grid1(S (i, j), k, n)* and the rest of *n-k* bits are randomly chosen from $r_1, r_2, \ldots, r_k$. The *n* bits are randomly reorganised and distributed to obtain *n* grids. The calculated theoretical contrast is 0.1666, 0.333, 0.5 for *t=2,3,4* in *(2, 4)* scheme and 0.0833, 0.1666, 0.25 for *t=3,4,5* in *(3, 5)* scheme.

Algorithm 4

Step 4.1 ∀ *S(i, j),* where $1 \le i \le M$, $1 \le j \le N$, repeat steps 4.2 and 4.3

Step 4.2 Generate $r_1, r_2, \ldots, r_k$ and repeat $\lfloor n/k \rfloor$ times to obtain $k\lfloor n/k \rfloor$ bits

$[r_1, r_2, \ldots r_k] = Random\_Grid1(S(i, j), k, n)$

Step 4.3 Generate $n - k\lfloor n/k \rfloor$ bits

$[r_{k+1}, r_{k+2}, \ldots, r_n] \in \{0, 1\}$

Step 4.4 Randomly rearrange and distribute *n* bits

$[S_1, S_2, \ldots, S_n] = Random\_Perm\ (r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$

In Algorithm 4[13], *k* bits are generated by the basic algorithm and repeat [*n/k*] times to obtain *k* * [*n/k*] bits. The rest of the *n*—*k* * [*n/k*] bits are arbitrarily chosen from normal distribution. These bits are shuffled and distributed to produce *n* shares. For $k > n/2$, this method works same as Chen & Tsao's method. For (*k, n*) scheme, when $k \le n/2$, this method improves the contrast of Chen & Tsao's scheme. The value of theoretical contrast is 0.1427, 0.2499, 0.2499 respectively for *t=2,3,4* in *(2, 4)* scheme and 0.0224, 0.0480, 0.0622 respectively for *t=4,5,6* in *(4, 6)* scheme.

Algorithm 5

Step 5.1 ∀ *S(i, j),* where $1 \le i \le M$, $1 \le j \le N$, repeat steps 5.2 and 5.3

Step 5.2 Generate $r_1, r_2, \ldots, r_k$

$[r_1, r_2, \ldots r_k] = Random\_Grid1(S(i, j), k, n)$

Step 5.3 Generate *n-k* bits

Set $r_{k+1} = r_1, r_{k+2} = r_2, \ldots, r_{2k} = r_k, r_{2k+1} = r_1$

$$r_n = \begin{cases} r_{(n \bmod k)} \\ r_k \text{ if } (n \bmod k) = 0 \end{cases}$$

Step 5.4 Randomly rearrange and distribute *n* bits

$[S_1, S_2, \ldots, S_n] = Random\_Perm\ (r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$

In Algorithm 5 [14], k bits are generated by *Random_Grid1(S (i, j), k, n)* and the remaining.

n—k bits are computed by assigning $r_{k+1}=r_1$, $r_{k+2}=r_2,\ldots,r_{2k}=r_k$, $r_{2k+1}=r_1$. These n bits are randomly reordered and allotted to n shares. The theoretical contrast is 0.2873, 0.5018, 0.5018 for $t=2,3,4$ and 0.0854, 0.1889, 0.2485 for $t=3,4,5$ in *(2, 4)* and *(3, 5)* schemes respectively.

Algorithm 6

Step 6.1 $\forall\, S(i,j)$, where $1 \le i \le M$, $1 \le j \le N$, repeat steps 6.2 to 6.6

Step 6.2 Generate $r_1, r_2, \ldots, r_k$

$$[r_1, r_2, \ldots r_k] = Random\_Grid1\ (S\ (i,j), k, n)$$

Step 6.3 Generate n-k bits

If $n > k$, set $r_{k+1}, r_{k+2}, \ldots, r_n \in \{0, 1\}$

Step 6.4 If $n > k$, go to step 6.5; else go to step 6.6

Step 6.5 If $S(i,j) = r_1 \oplus r_2 \oplus \ldots \oplus r_n$ go to step 6.7 else go to step 6.6

Step 6.6 Randomly choose $p \in \{k+1, k+2, \ldots, n\}$, then flip $r_p = \sim r_p$

Step 6.7 Randomly rearrange and distribute n bits

$$[S_1, S_2, \ldots, S_n] = Random\_Perm\ (r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$$

The theoretical contrast for the algorithm is given by the equation

$$\alpha = \begin{cases} \frac{2*tC_k}{(2^t+1)nC_k - tC_k}, & for\ k \le t < n\ or\ n = k \\ \frac{1}{2^{n-2}}, & for\ k < n\ and\ t = n \end{cases} \tag{9}$$

For *(3, 5)* $t = 3 \rightarrow \alpha = \frac{2*3C_3}{(2^3+1)5C_3 - 3C_3} = \frac{2}{89} = 0.0225$

$$t = 4 \rightarrow \alpha = \frac{2 * 4C_3}{(2^4+1)5C_3 - 4C_3} = \frac{8}{116} = 0.0482$$

$$t = 4 \rightarrow \alpha = \frac{1}{2^{n-2}} = \frac{1}{8} = 0.125$$

Algorithm 6 [15] generates k bits by *Random_Grid1*(*S (i, j), k, n*) and n—k bits are randomly generated by normal distribution if $n>k$. If $S(i,j)=r_1 \oplus r_2 \oplus \ldots \oplus r_n$, randomly rearrange the bits to get n shares or else randomly select p from $k+1, k+2, \ldots, n$

and then flip values of r_p to $\sim r_p$. Then rearrange and distribute the values to n shares. The value of theoretical contrast is 0.0689, 0.1176, 0.25 for $t=2,3,4$ in *(2, 4)* and 0.0225, 0.0482, 0.125 for $t=3,4,5$ in *(3, 5)* respectively.

Algorithm 7

Step 7.1 $\forall$ $S(i, j)$, where $1 \leq i \leq M$, $1 \leq j \leq N$, repeat steps 7.2 to 7.4

Step 7.2 Generate $r_1, r_2, \ldots, r_{k-1}$

$$[r_1, r_2, \ldots, r_{k-1}] \in \{0, 1\}$$

Step 7.3 Compute r_k

$$r_k = S(i, j) \oplus r_{k-1} \oplus \ldots \ldots \oplus r_1$$

Step 7.4 Generate n-k bits

$$\begin{cases} r_{k+1} = S(i, j) \oplus r_k \oplus r_{k-1} \oplus \ldots \ldots \oplus r_2 \\ r_{k+2} = S(i, j) \oplus r_{k+1} \oplus r_k \oplus \ldots \ldots \oplus r_3 \\ \\ r_n = S(i, j) \oplus r_{n-1} \oplus r_{n-2} \oplus \ldots \ldots \oplus r_{n-k+1} \end{cases}$$

Step 7.5 Randomly rearrange and distribute n bits

$$[S_1, S_2, \ldots, S_n] = Random\_Perm\,(r_1, r_2, \ldots r_k, r_{k+1}, r_{k+2}, \ldots, r_n)$$

The theoretical contrast for the algorithm is given by the equation

$$\alpha = \frac{Prob_k * (\frac{1}{2})^{k-1}}{1 + \sum_{w=1}^{k-1} Prob_w * (\frac{1}{2})^w} \text{ where } 1 \leq w \leq k \leq t \leq n \tag{10}$$

For *(3, 5)*, r_1, r_2, r_3, r_4, r_5 are split into three groups $GP_1=\{r_1, r_4\}$, $GP_2=\{r_2, r_5\}$, $GP_3=\{r_3\}$, where $r_1=r_4$ and $r_2=r_5$. For computing $Prob_w$ with $w=3$, $t=3$, the probability of getting t pixels from any of w groups determines the result. The permitted groups of chosen pixels are $\{r_1, r_2, r_3\}$, $\{r_1, r_5, r_3\}$, $\{r_4, r_5, r_3\}$, $\{r_4, r_5, r_3\}$ the total is 4. There are $5C_3=10$ possible combinations with 3 pixels. As a result, the value of $Prob_{w=3}$ is $4/10=2/5$. Similarly $Prob_{w=2}$ is $3/5$. Combining 3 shares, theoretical contrast, $\alpha = \frac{\left(\frac{2}{5}\right)^1 * \left(\frac{1}{2}\right)^2}{1+\left(\frac{3}{5}\right) * \left(\frac{1}{2}\right)^2} = \frac{2}{23} = 0.0870$

Algorithm 7 [16] generates *k—1* bits randomly from normal distribution and r_k by performing XOR operation with $S(i, j)$ and $r_1, r_2, \ldots r_{k-1}$. The rest of the *n-k* bits are generated by $r_{k+1} = S(i, j) \oplus r_k \oplus r_{k-1} \oplus \ldots \oplus r_2$ and $r_n = S(i, j) \oplus r_{n-1} \oplus r_{n-2} \oplus \ldots \oplus r_{n-k+1}$. Rearrange the *n* bits and then allotted to *n* shares. The theoretical contrast is 0.2857, 0.5, 0.5 for $t=2,3,4$ and 0.0870, 0.1905, 0.25 for $t=3,4,5$ in *(2, 4)* and *(3, 5)* schemes respectively.

Algorithm 8

Step 8.1 Construct the $2^{k-1} \times k$ basis matrix $B_0{}^{even}$ (even numbers of 1s) and $2^{k-1} \times k$ basis matrix $B_1{}^{odd}$ (odd numbers of 1s)

Step 8.2 $\forall$ $S(i, j)$, where $1 \le i \le M$, $1 \le j \le N$, repeat steps 8.2 to 8.4

Step 8.2.1 If $S(i, j) = 0$, arbitrarily choose a row R={ $\overline{r}$ } from $B_0{}^{even}$ where $1 \le g \le k$

Step 8.2.2 If $S(i, j) = 1$, arbitrarily choose a row R={ $\overline{r}$ } from $B_1{}^{odd}$ where $1 \le g \le k$

Step 8.3 Generate *n-k* grids

$$\overline{r_{k+1}} = \overline{r_{k+2}} = \ldots\ldots = \overline{r_n} = 0$$

Step 8.4 Randomly rearrange *n* bits

$$[r_1, r_2, \ldots, r_n] = Random\_Perm\left(\overline{r_1}, \overline{r_2}, \ldots\ldots, \overline{r_n}\right)$$

Step 8.5 Distribute *n* bits

$$[S_1, S_2, \ldots, S_n] = [r_1, r_2, \ldots, r_n]$$

The theoretical contrast for the algorithm is given by the equation

$$\alpha = \frac{\dfrac{\binom{n-k}{t-k}}{\binom{n}{t}} * \dfrac{1}{2^{k-1}}}{1 + \sum_{max\{0,t+k-n\} \le g < k} \dfrac{\binom{k}{g}\binom{n-k}{t-g}}{\binom{n}{t}} * \dfrac{1}{2^g}} \tag{11}$$

Table 2 Theoretical values of (2, 4) and (3, 5) VCS

Sl. no	Author and year	Theoretical contrast(α)–OR					
		(2,4)			(3,5)		
		$t=2$	$t=3$	$t=4$	$t=3$	$t=4$	$t=5$
1	Chen & Tsao, [9]	0.0690	0.1176	0.125	0.0225	0.0482	0.0625
2	Wu & Sun, [11]	0.2	0.3333	0.5	0.0625	0.1364	0.25
3	Lee et al., [12]	0.1666	0.3333	0.5	0.0833	0.1666	0.25
4	Guo et al., [13]	0.1427	0.2499	0.2499	0.0224	0.0480	0.0622
5	Liu et al., [14]	0.2873	0.5018	0.5018	0.0854	0.1889	0.2485
6	Yan et al., [15]	0.0689	0.1176	0.25	0.0225	0.0482	0.125
7	Hu et al., [16]	0.2857	0.5	0.5	0.0870	0.1905	0.25
8	Zhao & Fu, [17]	0.0555	0.2	0.5	0.0192	0.0870	0.25

$$\text{For } (3,5),\ t = 3 \rightarrow \alpha = \frac{\frac{\binom{2}{0}}{\binom{5}{3}} * \frac{1}{2^2}}{1 + \sum \frac{\binom{3}{1}\binom{2}{2}}{\binom{5}{3}} * \frac{1}{2^1} + \frac{\binom{3}{2}\binom{2}{1}}{\binom{5}{3}} * \frac{1}{2^2}} = \frac{1}{52} = 0.0192$$

Table 3 PSNR & SSIM values for (3, 5) VCS

Sl.no	Author and year	Experimental contrast (τ)–OR					
		PSNR			SSIM		
		$t=3$	$t=4$	$t=5$	$t=3$	$t=4$	$t=5$
1	Chen & Tsao, [9]	50.9691	51.1350	51.2022	0.1716	0.2548	0.3339
2	Wu & Sun, [11]	51.5224	51.9072	52.4723	0.1108	0.2465	0.5097
3	Lee et al., [12]	47.2899	46.6619	46.0034	0.0021	0.0097	0.0217
4	Guo et al., [13]	46.0149	45.4370	45.0582	0.0030	0.0445	0.0811
5	Liu et al., [14]	47.2171	46.6055	45.9724	0.0015	0.0089	0.0213
6	Yan et al., [15]	47.0149	46.6349	46.3208	0.0199	0.0359	0.0443
7	Hu et al., [16]	51.4363	51.4354	52.4558	0.3517	0.4992	0.5084
8	Zhao & Fu, [17]	51.4386	51.7282	52.5806	0.0501	0.1566	0.4725

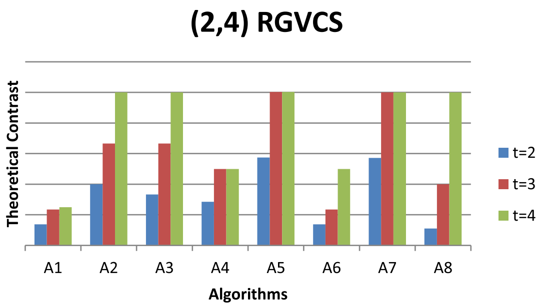


Fig. 2 Theoretical Contrast–*(2, 4) VCS*

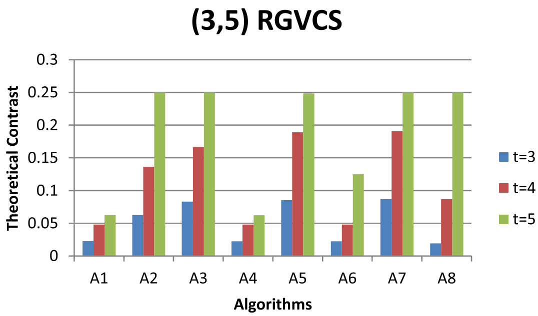


Fig. 3 Theoretical Contrast–*(3, 5) VCS*

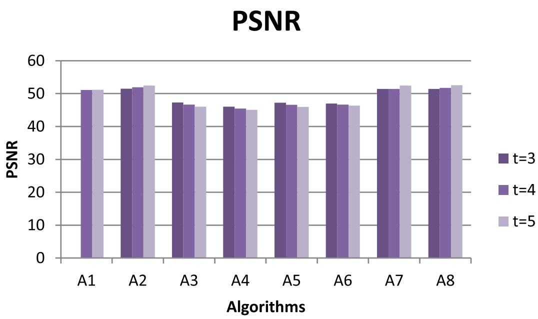


Fig. 4 PSNR values

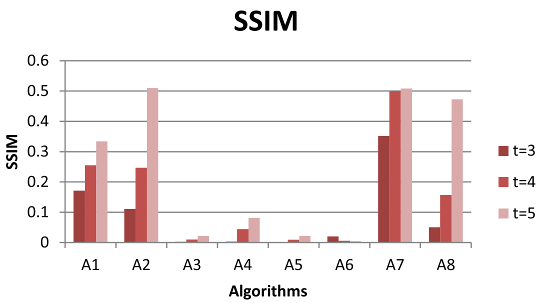


Fig. 5 SSIM values

Fig. 6 Secret image

$$t = 4 \rightarrow \alpha = \frac{\frac{\binom{2}{1}}{\binom{5}{4}} * \frac{1}{2^2}}{1 + \frac{\binom{3}{2}\binom{2}{2}}{\binom{5}{4}} * \frac{1}{2^2}} = \frac{2}{23} = 0.0870$$

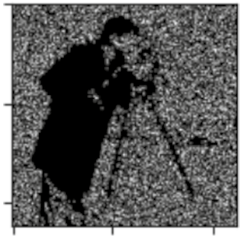
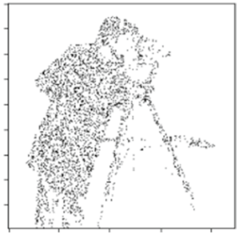
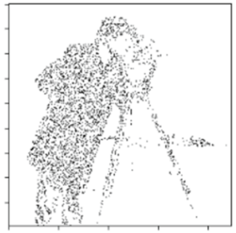
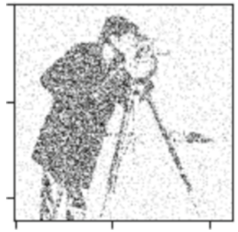
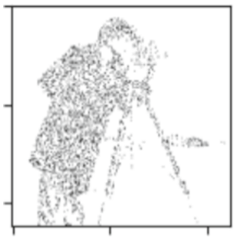
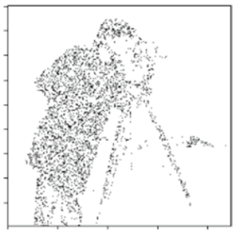
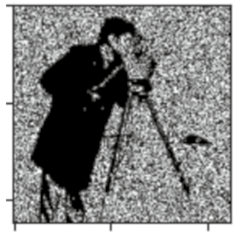
$$t = 5 \rightarrow \alpha = \frac{\frac{\binom{2}{2}}{\binom{5}{5}} * \frac{1}{2^2}}{1} = \frac{1}{4} = 0.25$$

In Algorithm 8 [17], $2^{k-1} \times k$ basis matrices B_0^{even} and B_1^{odd} are created. Rows are selected randomly from B_0^{even} and B_1^{odd} according to the value of each pixel and hence k bits are generated. Then generate remaining *n-k* bits. Randomly rearrange and distribute these bits to generate n shares. The theoretical contrast is 0.0555, 0.2, 0.5 for $t=2,3,4$ respectively in *(2, 4)* scheme and 0.0192, 0.0870, 0.25 for $t=3,4,5$ respectively in *(3, 5)* scheme.

5 Experimental findings and discussions

To determine the contrast and security of the compared algorithms, experiments and analyses were conducted. The secret image can be either partially or fully reconstructed by overlapping at least k or more shares. Meaningless images are obtained when less than k numbers of shares are overlapped. The images supporting the experiments are taken from USC-SIPI image dataset. Theoretical contrast (τ), experimental contrast (α) and visual quality are compared to evaluate the contrast

Table 4 Comparison of evaluation metrics of reconstructed image in related schemes

Guo et al. [13]	Lee et al. [12]	Wu & Sun [11]	Chen & Tsao [9]
$\alpha = 0.0622$ $\tau = 0.0811$	$\alpha = 0.25$ $\tau = 0.0217$	$\alpha = 0.25$ $\tau = 0.5097$	$\alpha = 0.0625$ $\tau = 0.3339$
			
Zhao & Fu [17]	Hu et al. [16]	Yan et al. [15]	Liu et al. [14]
$\alpha = 0.25$ $\tau = 0.4725$	$\alpha = 0.25$ $\tau = 0.5084$	$\alpha = 0.125$ $\tau = 0.0443$	$\alpha = 0.2485$ $\tau = 0.0213$
			

and similarity index of original and reconstructed images. The algorithms are implemented using Python and hence experimental contrast is calculated for the various schemes. The results of theoretical contrast of OR based *(2, 4)* and *(3, 5)* schemes by Algorithms 1–8 is depicted in Table 2. The stacked results of different combinations such as $t=2,3,4$ and $t=3,4,5$ are also shown in the table.

In 2016, Yan et al. [18] studied and demonstrated the interval of values corresponding to contrast by conducting several experiments. By studying and evaluating the outputs, we found that the visual quality of the retrieved images in different algorithms varies according to the values of k and t. It is understood that the retrieved image can be identified as the original image when the calculated value of α is greater than zero. However, it is hard to recognize with the HVS when α is less than a specific value. The results of PSNR and SSIM for *(3, 5)* scheme for the values of $t=3,4,5$ are given in Table 3.

Figures 2, 3, 4, 5 shows the values of theoretical contrast and experimental contrast of various *(k, n)* algorithms. From the depicted graphs, it can be understood that Hu et al. [16] performs better in terms of contrast and for the values of PSNR and SSIM. The secret image is given in Fig. 6. Table 4 shows the comparative analysis of visual quality, theoretical and experimental contrasts. Based on the above comparisons, the visual quality of retrieved secret images in Chen & Tsao [7] and Guo et al. [13] are less, as there are more number of black pixels in the retrieved image. Besides, Wu & Sun [11] and Hu et al. [16] schemes have enhanced visual quality.

6 Conclusion and future enhancements

In this paper, we mainly focused on various schemes of *(k, n)* VCS based on random grids. It has been found out that Hu et al.'s algorithm surpasses other related algorithms in terms of visual quality, theoretical and experimental contrast. Some schemes used improved algorithms in order to enhance the existing contrast. General formula for calculating theoretical contrast is provided. It is also found that out of the n pixels, first k pixels are associated with the contrast and therefore the last n-k pixels can be generated by making use of the first k pixels. The relationship between visual quality and contrast deserves further investigation. The contrast of the schemes which are not given in terms of k, t and n can be considered as an open issue which is to be addressed in the future.

Author contributions Authors 1 & 2 equally contributed to the manuscript and Author 3 supervised the work. All authors reviewed the manuscript.

Data availability The images supporting the study can be found in USC-SIPI Image Database.

Declarations

Conflict of interest The authors state that they have no financial/non-financial interests.

Ethical approval Consent to participate has to be considered is not applicable.

References

1. Naor, M., & Shamir, A. (1995). Visual cryptography. In *Advances in Cryptology—EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings 13* (pp. 1–12). Springer Berlin Heidelberg.
2. Shyu SJ (2006) Efficient visual secret sharing scheme for color images. Pattern Recogn 39(5):866–880
3. Shyu SJ (2009) Image encryption by multiple random grids. Pattern Recogn 42(7):1582–1596
4. Chen TH, Tsao KH, Lee YS (2012) Yet another multiple-image encryption by rotating random grids. Signal Process 92(9):2229–2237
5. Kafri O, Keren E (1987) Encryption of pictures and shapes by random grids. Opt Lett 12(6):377–379
6. Francis N, Monoth T (2023) Security enhanced random grid visual cryptography scheme using master share and embedding method. Int J of Inf Technol 15(7):1–7
7. Chen TH, Tsao KH (2009) Visual secret sharing by random grids revisited. Pattern Recogn 42(9):2203–2217
8. Shyu SJ (2007) Image encryption by random grids. Pattern Recogn 40(3):1014–1031
9. Chen TH, Tsao KH (2011) Threshold visual secret sharing by random grids. J Syst Softw 84(7):1197–1208
10. Chen, T. H., & Tsao, K. H. (2008). Image encryption by (n, n) random grids. In: Proceedings of 18th Information Security Conference, Hualien.
11. Wu X, Sun W (2013) Improving the visual quality of random grid-based visual secret sharing. Signal Process 93(5):977–995
12. Lee YS, Wang BJ, Chen TH (2013) Quality-improved threshold visual secret sharing scheme by random grids. IET Image Proc 7(2):137–143
13. Guo T, Liu F, Wu C (2013) Threshold visual secret sharing by random grids with improved contrast. J Syst Softw 86(8):2094–2109
14. Yan X, Liu X, Yang CN (2018) An enhanced threshold visual secret sharing based on random grids. J Real-Time Image Proc 14:61–73
15. Yan X, Wang S, Niu X, Yang CN (2015) Random grid-based visual secret sharing with multiple decryptions. J Vis Commun Image Represent 26:94–104
16. Hu H, Shen G, Fu Z, Yu B (2018) Improved Contrast for Threshold Random-grid-based Visual Cryptography. KSII Trans on Int Inf Systems 12(7):3401–3420
17. Zhao Y, Fu FW (2022) A contrast improved OR and XOR based (k, n) visual cryptography scheme without pixel expansion. J Vis Commun Image Represent 82:103408
18. Yan, X., Lu, Y., Huang, H., Liu, L., & Wan, S. (2016). Clarity corresponding to contrast in visual cryptography. In: Social Computing: Second International Conference of Young Computer Scientists, Engineers and Educators, ICYCSEE 2016, Harbin, China, August 20-22, 2016, Proceedings, Part I 2 (pp. 249-257), Springer, Singapore