Multimedia Tools and Applications (2023) 82:41987–42019
https://doi.org/10.1007/s11042-023-14998-7

Proactive visual cryptographic schemes for general access structures

Praveen K[1] · Sabyasachi Dutta[2] · Avishek Adhikari[3] · Sethumadhavan M[1]

Received: 2 March 2022 / Revised: 13 August 2022 / Accepted: 22 February 2023 /
Published online: 4 April 2023

Abstract

In the modern digital era, long-term protection of digital data, stored in a server, is essential, especially when it is in the format of medical images like electronic health records, MRI scans, etc. Visual cryptography is an efficient method to protect secret image data by encoding it into shares and keeping in different servers. However, if the shares, stored in several servers, remain unaltered for a long time, an adversary may capture the shares one by one and in the end, break the confidentiality of the secret data. In the current paper, we propose, to the best of our knowledge, the first proactive visual cryptographic scheme for general access structures. In our scheme, the shares are embedded within meaningful cover images and are stored in different servers. The main advantage of our scheme is that the meaningful share images are updated periodically without changing the original secret image. The updating/renewal procedure makes the previous shares statistically independent from the current shares. The secret image can be reconstructed only by the lastly updated share images. This technique prevents the aforementioned attack. The renewal procedure can be performed an unlimited number of times still keeping the quality of the reconstructed image unchanged. The mathematical analyses along with the experimental results exhibit the practicality of our proposed scheme.

✉ Praveen K
k_praveen@cb.amrita.edu

Sabyasachi Dutta
saby.math@gmail.com

Avishek Adhikari
avishek.adh@gmail.com

Sethumadhavan M
m_sethu@cb.amrita.edu

[1] TIFAC-CORE in Cyber Security Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India

[2] University of Calgary, Alberta, Canada

[3] Department of Mathematics, Presidency University, Kolkata, India

Keywords Visual cryptography · Extended visual cryptography · Proactive secret sharing · General access structure

1 Introduction

In the modern era, steganography and cryptography play very important roles as main instruments for data hiding techniques. Particularly, they have found important applications to protect healthcare data [19], biometric data [7], genomic data [18] and medical/biological image data [3, 4, 32, 34]. On the other hand, any of the information hiding techniques and machine learning framework like generative adversarial network(GAN) can be used in a blended way [36, 38] for securing images.

A visual cryptographic scheme (VCS), on a set of n participating servers, enables a user to encode a secret image into n shares and distribute these shares among n servers. Typically, the user then deletes the image from her storage, and later she contacts the servers when she wants to recover the image. The generation of these shares is done in such a manner so that only pre-specified "qualified" subsets of servers can be contacted to recover the image. On the other hand, the shares held by any subset which are not *qualified* can not reveal any information about the secret image. These latter type of subsets are known as "forbidden". Such a collection of qualified sets and forbidden sets constitute an *access structure* $\mathcal{A}$ on the set of n servers. In this paper, we consider access structures where any subset of servers is either a qualified set or a forbidden set. Most commonly known access structures are (k, n)-threshold access structures where any set of shares (of size at least k) are enough to reconstruct the secret, whereas, no collection of at most $k-1$ shares reveal any information about the secret.

Naor and Shamir [24] introduced and formalized the notion of visual secret sharing (a.k.a visual cryptography) and provided a methodology that did not require any computer participation for the recovery of secret images. This was made possible by using Boolean operation "OR" during the reconstruction process. "OR" based visual cryptography was later extended in [1, 2, 5, 11, 28] to general access structures and in [10, 12, 37, 40] to generalized OR-based color visual cryptographic schemes. Most of the constructions are realized by constructing so-called basis matrices. The major problems for any OR-based visual cryptographic scheme are the huge share size (pixel expansion) and very poor contrast of the reconstructed image.

To improve upon the quality (contrast) of the superimposed image, several attempts were made. Tuyls et al. [35] put forward a new model of VCS based on the polarization of light where the underlying mathematical operation was the Boolean "XOR" operation. In [35] the authors constructed a XOR-based (n, n)-VCS and proved that a XOR-based $(2, n)$-VCS is equivalent to a binary code. For further reading on the topic the reader may refer to [21, 41, 42]. All these papers have the common property that all of them are non-monotonic in nature, i.e., a superset of the minimal qualified set may not get the secret back. In the case of Liu et al. XOR-based step construction for general access structures [21], the pixel expansion is less when compared to other constructions in the literature. Several papers [17, 26, 29], used Liu et al.'s step based construction for general access structures [21] as building block.

Other models of sharing secret images include polynomial based image sharing schemes [14, 30, 33, 43] and the Chinese Remainder Theorem (CRT) based schemes [8, 20, 22, 39].

In this paper, we consider "XOR" operation based visual cryptographic schemes only.

1.1 Motivation & objective

Protecting data for long term. Data that are stored digitally sometimes require protection throughout their whole lifetime which may be varying depending on the nature of data and could be very long. One of the most important protection goals is *confidentiality* which means that only authorized servers/users can access the data and nobody else. The example of storing health-related data is an important case in point. The data must be securely stored at least as long the patients are alive. Another important issue in the context of such long-term storage is the issue of *availability* of the data *i.e.* the data can be retrieved at any point of time by legitimate users. Information theoretically secure secret sharing provides one methodology towards creating a system that can provide above mentioned "security" properties.

In a k-out-of-n secret sharing scheme, any k shares are enough to reveal/reconstruct the secret. If the shares are stored for a relatively long period, then it is only a matter of time that an adversary breaks into a sufficient number of servers (in this case, k many servers) and reconstructs the secret. In this case, the confidentiality of the secret data is lost if they are left alone for a long period. Such an adversary is called a `mobile` adversary who captures servers gradually over a period of time. The mobile adversary setting was originally presented in the context of secure systems by Ostrovsky and Yung [25].

One of the most important known approaches to address long-term confidentiality of secret data is to use *proactive secret sharing* (PSS) which was introduced by Herzberg et al. [16]. To prevent a `mobile` adversary from being able to collect enough shares over time to reconstruct the data, the shares are renewed periodically. After every renewal process, the servers stores the new shares which are generated as the outcome of the renewal process, and *deletes* the old shares. Note that, the renewal process must ensure that the shares before the renewal and after the renewal must be statistically independent; otherwise they may reveal nontrivial information about the secret entity.

Proactive secret sharing generally is built on secret sharing schemes which are linear - *i.e.* the secret is a linear combination/sum of chosen shares. Such linear secret sharing schemes exist and in fact, the first secret sharing scheme of Shamir [31] is linear. This immediately effects an inclusion of update/renewal procedure of the shares – every server or server independently generates shares of the value 0 in accordance with the access structure and send those values to every other server; the servers then update their existing share by adding the values which they receive from others. Since during the renewal process every server chooses 0 and secret shares the value among all the servers, the procedure along with the linearity property of underlying secret reconstruction together imply that the original secret remains the same but the shares are updated.

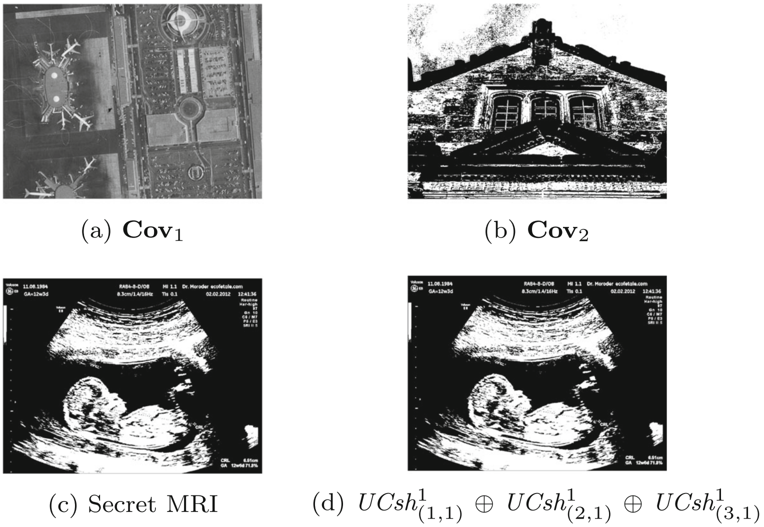
We refer to the share renewal process of Herzberg et al. [16] when the servers behave semi-honestly during the share update protocol. The process can further be modified to resist attacks from *malicious* servers which have been taken care of by using a verifiable secret sharing technique. For more details, we refer to the work of [16].

Related works on secret image data. Although there exist a plethora of works to ensure proactive security for secret sharing schemes and their applications, not much research has been carried out in the context of secret image data sharing which has its own challenges and applications. To the best of our knowledge, the only works which have addressed the issue are by Guo et al. [15] and Trujillo et al. [13]. The work of Guo et al. [15] is based on a proactive linear integer secret sharing scheme proposed by Ma and Ding [23]. They have meaningful shadows with reasonable embedding capacity and the secret image can be reconstructed losslessly. One drawback of their proposal is that the share renewal procedure could be performed only a limited number of times. Trujillo et al. [13] considered security against state- of-the-art machine learning algorithms to break the steganographic security and proposed scheme which is resilient to such attacks.

Challenges for VCS. The area of secret image sharing is closely related to the area of general secret sharing. On the other hand, visual cryptography follows a model concretely defined by [24] which differs from the security model of general secret sharing. Therefore, the constructions and the realizations of VCS differ significantly from those of Secret Image sharing. It introduces certain challenges to achieve the proactive nature of VCS and thus suitable modifications are required – they cannot be achieved directly from the constructions of [13, 15]. To point out one of the main difficulties, we refer to the basic constructions of XOR-based schemes [9, 35] etc. where the shares are generated by constructing so-called "basis matrices" where random column permutations are applied on the columns and each row is given as the shares to the servers. It is not very hard to see that such a construction cannot be used to achieve proactivity because the servers then must know the exact column permutation in order to introduce proactive nature in the scheme. However, revealing the permutation seriously affects the security of the scheme. We can only bypass the difficulty if we give multiple shares to the servers for each secret pixel.

Applications to storing medical images. The patient's confidentiality regarding his/her treatment is of vital importance and should be protected. It is the right of an individual that his/her personal and medical information is kept confidential. Individuals rely upon Cloud Storage Providers (CSP) for storing their information due to the lack of storage space in their personal gadgets. The main concern in cloud storage is its confidentiality. To obtain confidentiality users will store the secret image in the form of secret shares in various CSP and reconstruct the secret image back by combining the shares [27].

Figure 1 shows two cover images, secret MRI image and reconstructed MRI image for the qualified subset ($\{\mathsf{Ser}_1, \mathsf{Ser}_2, \mathsf{Ser}_3\}$). Figure 2 shows the experimental results of generated meaningful cover shares of the participants in the qualified subsets ($\{\mathsf{Ser}_1, \mathsf{Ser}_2, \mathsf{Ser}_3\}$



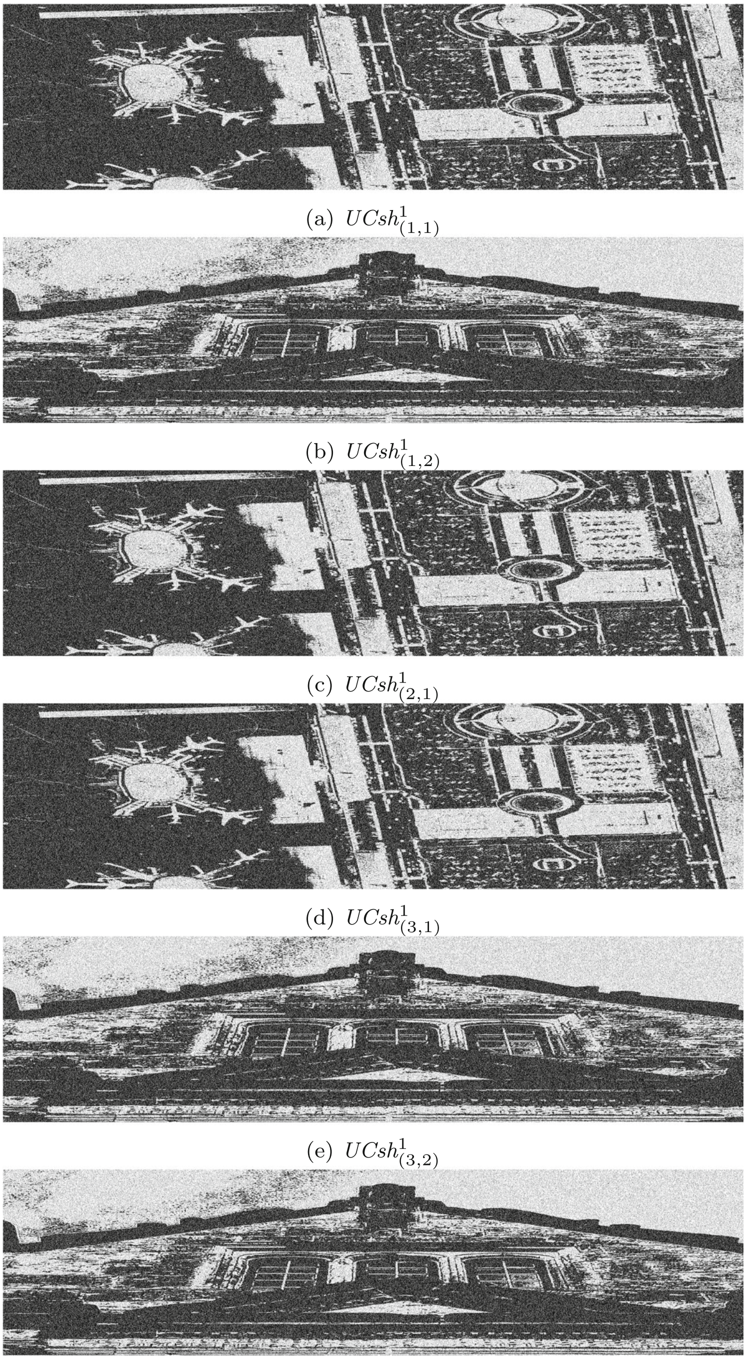
(a) $\mathbf{Cov}_1$

(b) $\mathbf{Cov}_2$

(c) Secret MRI

(d) $UCsh^1_{(1,1)} \oplus UCsh^1_{(2,1)} \oplus UCsh^1_{(3,1)}$

Fig. 1 Experimental results on **PS-2(DC)** (Section 4.2) for access structure$\Gamma = \{\{\mathsf{Ser}_1, \mathsf{Ser}_2, \mathsf{Ser}_3\}, \{\mathsf{Ser}_1, \mathsf{Ser}_2, \mathsf{Ser}_4\}, \{\mathsf{Ser}_1, \mathsf{Ser}_3, \mathsf{Ser}_4\}, \{\mathsf{Ser}_2, \mathsf{Ser}_3, \mathsf{Ser}_4\}\}$: (a) Cover Image 1, (b) Cover Image 2, (c) Secret MRI, (d) Reconstructed output after first share renewal procedure



(a) $UCsh^1_{(1,1)}$

(b) $UCsh^1_{(1,2)}$

(c) $UCsh^1_{(2,1)}$

(d) $UCsh^1_{(3,1)}$

(e) $UCsh^1_{(3,2)}$

(f) $UCsh^1_{(4,2)}$

Fig. 2 (a-f) Cover shares of servers Ser_1, Ser_2, Ser_3, Ser_4 after first renewal using **PS-2(DC)** for access structure $\Gamma = \{\{\mathsf{Ser}_1, \mathsf{Ser}_2, \mathsf{Ser}_3\}, \{\mathsf{Ser}_1, \mathsf{Ser}_2, \mathsf{Ser}_4\}, \{\mathsf{Ser}_1, \mathsf{Ser}_3, \mathsf{Ser}_4\}, \{\mathsf{Ser}_2, \mathsf{Ser}_3, \mathsf{Ser}_4\}\}$

and $\{\mathsf{Ser}_1, \mathsf{Ser}_3, \mathsf{Ser}_4\}$) of Γ, after "proactively" renewing the shares of secret MRI image. Detailed explanation for this experiment is given in Section 4.2.

1.2 Our contribution

In this work, we put forward a proactive visual cryptographic scheme for general access structures. To the best of our knowledge, this is the first work on a proactive visual cryptographic scheme. We take "XOR" based constructions in order to include share update/renewal procedure into the system. We have taken a generic approach where a XOR-based visual cryptographic scheme for general access structures can be upgraded to achieve proactive security. In particular, we have considered two methodologies of constructing underlying basic XOR-based VCS. The first one uses linear algebraic technique on a finite field of size 2 and the second one is using a step construction technique which partitions the access structure and then uses (2, 2)- sharing protocol iteratively till all the servers are exhausted. There is no limit to the number of renewal protocols performed by the servers. The quality of the reconstructed secret image remains the same all throughout the lifetime of the secret image. One of the important features in our work that makes it more applicable in practice is the lossless recovery of secret images. Table 1 shows the comparison of our proposed schemes with related works.

We have provided experimental analysis on the pixel expansion and the quality of the reconstructed image. In the later part of the paper, we have used cover images as in steganography to spawn meaningful shares on top of "proactive" ness of the schemes. As already mentioned previously, this work has one very important application in the field of storing medical images.

2 Preliminaries

In the following, we define the basic terminologies required for the rest of the paper. Table 2 shows the notations used in the paper with its corresponding descriptions.

2.1 XOR based visual cryptographic scheme (XVCS)

Let S be an $n \times m$ Boolean matrix and let $X \subset \mathcal{P}$. By $S[X]$ we denote the matrix obtained by restricting the rows of S to the indices belonging to X. Further, for any $X \subset \mathcal{P}$ the vector obtained by Implementing the Boolean XOR operation "$\oplus$", to the rows of $S[X]$ is denoted

Table 1 Comparison with other schemes

Parameters	Guo et al. [15]	Trujillo et al. [13]	Our
Operations	Field mult. addition	Field mult. addition	Boolean XOR, AND
Access structure	Threshold	Threshold	General
Renewal Process	Limited (≤ 10)	–	Unlimited times
Meaningful shares	Yes	Yes	Yes
Reconstructed image quality	Degrades after few iterations	–	Lossless recovery

Table 2 Notations

Notation	Description
$\bigoplus$	Bitwise Boolean Exclusive OR operation
$\bigodot$	Bitwise Boolean AND operation
$\lvert P \rvert$	Number of elements in the set P
Γ	Access structure
Γ_{QM}	Collection of Minimal Qualified Sets
Γ_{FM}	Collection of Maximal Forbidden Set
α	Relative Contrast
$\vert\vert$	Concatenation
$w(S_X)$	Hamming weight of vector S_X
PE and APE	Pixel expansion and Average pixel expansion
SI and Cov/CI_n	Secret Image and Cover Image/n^{th} cover image
Ecov	Pixel expanded Cover Image
XVCS	XOR based VCS
PS-1 and PS-2	Proactive scheme 1 and 2 with random shares
PS-2(**SC**)	PS-2 with meaningful shares generated from **s**ame **c**over images
PS-2(**DC**)	PS-2 with meaningful shares generated from **d**ifferent **c**over images
$sh_{(u,j)}$	j^{th} random share of u^{th} server
$sh^t_{(u,k,j)[0]}$	j^{th} random share of k^{th} server received from u^{th} server after t^{th} iteration
$Ush^t_{(u,j)}$	j^{th} random share of u^{th} server after t^{th} iteration
$Csh_{(u,j)}$	j^{th} meaningful share of u^{th} server
$UCsh^t_{(u,j)}$	j^{th} meaningful share of u^{th} server after t^{th} iteration

by S_X. The Hamming weight of the row vector which represents the number of ones in the vector S_X is denoted by $w(S_X)$.

Definition 1 Let $\mathcal{P} = \{1, 2, 3, \ldots, n\}$ be a set of servers. An access structure on $\mathcal{P}$ is defined by a tuple $\Gamma = (\Gamma_{QM}, \Gamma_{FM})$ such that $\Gamma_{QM} \cap \Gamma_{FM} = \emptyset$. Γ_{QM} denotes a collection of subsets of $\mathcal{P}$ and each subset is called a minimal qualified set. On the other hand, Γ_{FM} is known as the collection of maximal forbidden subsets of $\mathcal{P}$.

Definition 2 Let $\mathcal{P} = \{1, 2, 3, \ldots, n\}$ be a set of servers and Γ be an access structure defined on $\mathcal{P}$. A XOR-based visual cryptographic scheme (XVCS) on $\mathcal{P}$ satisfies the following two conditions:

1. Any minimal qualified set of servers can recover the secret.
2. Any forbidden set of servers does not have any information about the secret image.

In the following, we give a standard definition of an XVCS through basis matrices.

Definition 3 (Basis Matrices [9, 35]) An XVCS on an access structure Γ is realized using two $n \times m$ binary matrices S^0 and S^1 called basis matrices, if there exist two sets of non-negative real numbers $\{\alpha_X\}_{X\in\Gamma_{QM}}$ and $\{t_X\}_{X\in\Gamma_{QM}}$ such that the following two conditions hold:

1. (*contrast*) If $X \in \Gamma_{QM}$, then S^0_X, the "XOR" of the rows indexed by X of S^0, satisfies $w(S^0_X) \leq t_X - \alpha_X \cdot m$; whereas, for S^1 it results in $w(S^1_X) \geq t_X$.
2. (*security*) If $Y = \{i_1, i_2, \ldots, i_s\} \in \Gamma_{FM}$ then the two $s \times m$ matrices $S^0[Y]$ and $S^1[Y]$ obtained by restricting S^0 and S^1 respectively to rows $i_1, i_2, \ldots, i_s$ are identical up to a column permutation.

The number m is called the pixel expansion of the scheme. Also α_X and $\alpha_X \cdot m$ respectively denote the relative contrast and contrast of the recovered image reconstructed by the minimal qualified set X.

Definition 4 (Collection of Matrices [9, 35]) Let $\mathcal{P} = \{1, 2, 3, \ldots, n\}$ be a set of servers. Let $\Gamma = (\Gamma_{QM}, \Gamma_{FM})$ be an access structure defined on $\mathcal{P}$. Let m and $\{h_X\}_{X \in \Gamma_{QM}}$ be non-negative integers satisfying $1 \leq h_X \leq m$. Two collections of $n \times m$ binary matrices $\mathcal{C}_0$ and $\mathcal{C}_1$ realizes XVCS on Γ, if there exists $\{\alpha_X > 0 : X \in \Gamma_{QM}\}$ such that

1. For any $S \in \mathcal{C}_0$, the "XOR" operation of the rows of $S[X]$ for any minimal qualified set $X \in \Gamma_{QM}$ results in a vector v_0 satisfying $w(v_0) \leq h_X - \alpha_X \cdot m$.
2. For any $T \in \mathcal{C}_1$, the "XOR" operation of the rows of $T[X]$ for any minimal qualified set X results in a vector v_1 satisfying $w(v_1) \geq h_X$.
3. Any forbidden set $Y \in \Gamma_{FM}$ has no information on the shared image. Formally, the two collections of $|Y| \times m$ matrices D_t, with $t \in \{0, 1\}$, obtained by restricting each $n \times m$ matrix in C_t to rows indexed by Y are indistinguishable in the sense that they contain the same matrices with the same frequencies.

2.2 Linear Algebraic Construction of XVCS [9]

Dutta and Adhikari [9] provided an efficient linear algebra based construction of XVCS realizing any (k, n) threshold access structure. In their construction method, the $\binom{n}{k}$ many minimal qualified sets $B_1, \ldots, B_q$ (where $q = \binom{n}{k}$) are partitioned into $\{(B_1, B_2), (B_3, B_4), \ldots, (B_{q-1}, B_q)\}$ if q is even or $\{(B_1, B_2), (B_3, B_4), \ldots, (B_{q-2}, B_{q-1}), (B_q)\}$ if q is odd.
For each participant i, assign a variable x_i. For each size 2 subset (B_{r-1}, B_r) consider the following systems of equations

$$\left.\begin{array}{rl} \sum\limits_{s \in B_{r-1}} x_s = & 0(mod\ 2) \\ \sum\limits_{t \in B_r} x_t = & 0(mod\ 2) \\ (x_u)_{u \notin B_{r-1} \cup B_r} & = \mathbf{0} \end{array}\right\} \cdots (r) \quad \text{and} \quad \left.\begin{array}{rl} \sum\limits_{s \in B_{r-1}} x_s = & 1(mod\ 2) \\ \sum\limits_{t \in B_r} x_t = & 1(mod\ 2) \\ (x_u)_{u \notin B_{r-1} \cup B_r} & = \mathbf{0} \end{array}\right\} \cdots (r')$$

If q is odd then the last system is defined as $\sum\limits_{s \in B_q} x_s = 0(mod\ 2)$ (or $1(mod\ 2)$) along with $(x_u)_{u \notin B_q} = \mathbf{0}$ S^0_r denote the Boolean matrix whose columns are all possible solutions of the system (r). Also, let S^1_r denote the Boolean matrix whose columns are all possible solutions of the system (r'). Let (S^0, S^1) denote the pair of Boolean matrices obtained by the concatenations: S^0=$S^0_2||S^0_4|| \cdots ||S^0_{\lceil \frac{q}{2} \rceil}$ and S^1=$S^1_2||S^1_4|| \cdots ||S^1_{\lceil \frac{q}{2} \rceil}$. It was shown in [9] that the pair (S^0, S^1) obtained in the above manner admits basis matrices realizing k-out-of-n XVCS.

2.3 Step construction of XVCS by Liu et al. [21]

Let $C_0 = \left\{ \left\{ \begin{matrix} 1 \\ 1 \end{matrix} \right\}, \left\{ \begin{matrix} 0 \\ 0 \end{matrix} \right\} \right\}$ and $C_1 = \left\{ \left\{ \begin{matrix} 1 \\ 0 \end{matrix} \right\}, \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\} \right\}$ be the two collections of matrices used for sharing a 0 and 1 pixel respectively in XOR based $(2, 2)$-VCS. Here the reconstructed image *RI* is same as the secret image *SI*. Liu et al. [21] in 2010 developed a non-monotonic step construction by recursively calling XOR based (2,2)-VCS. In step construction, each server holds different number of shares, so that the average pixel expansion (APE) was introduced instead of pixel expansion. The APE [21] is defined as the average value of the total pixel expansions of the share images that each server holds. The APE and relative contrast of step construction is better when compared to other results and is given in Table 1 of paper [17]. Step construction was developed based on the following definitions and theorem.

Definition 5 (Equivalent servers [21]) Let $(\Gamma_{QM}, \Gamma_{FM})$ be an access structure on P. If servers p_i and p_j satisfy that, for all $A \in \Gamma_{FM}$, $p_i \in A$ if and only if $p_j \in A$, then server p_i and p_j are considered to be equivalent servers on Γ_{QM}, denoted by $p_i \sim p_j$.

The relation $\sim$ is an equivalence relation on P. Let $\widetilde{P}$ be the quotient set derived from P based on $\sim$. The following definition shows how to simplify the access structure based on equivalent servers.

Definition 6 (Simplifying access structure [21]) The simplified access structure based on the equivalent servers is $\widetilde{\Gamma_{QM}} = \{\widetilde{A}\colon A \in \Gamma_{QM}\}$, where the set $\widetilde{A} = \{\tilde{p_i} \in \widetilde{P}\colon p_i \in A\}$ is called the corresponding set of A and $\widetilde{p_i}$ is called the equivalence class of p_i. Γ_{QM} is called the most simplified access structure when $\widetilde{\Gamma_{QM}} = \Gamma_{QM}$.

Theorem 1 [21] *Let $\widetilde{\Gamma_{QM}} = \{\widetilde{A}\colon A \in \Gamma_{QM}\}$. By distributing the same share to the equivalent servers, any construction of VCS for the $\widetilde{\Gamma_{QM}}$ is also a construction of VCS for the Γ_{QM}.*

For a detailed description of the share generation procedure for any access structure, we refer to the paper by Liu et al. [21]. The basic idea is to divide/partition the given access structure into simpler sub-access structures and apply $(2, 2)$-XVCS iteratively for each simplified sub-access structure to generate the shares for the servers. In the following, we describe the main steps of the share generation algorithm through Example 1.

Example 1 Let $\mathsf{Ser} = \{Ser_1, Ser_2, Ser_3, Ser_4\}$ and $\text{SI} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ denote the set of servers and secret image respectively. Let $\Gamma_{QM} = \{\{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}, \{Ser_2, Ser_3, Ser_4\}\}$ be the collection of minimal qualified sets. The step construction by Liu et al. [21] for XVCS divides Γ_{QM} into two parts $\Gamma_1 = \{\{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}\}$ and $\Gamma_2 = \{\{Ser_2, Ser_3, Ser_4\}\}$. Then dealer performs the following steps.
1) For Γ_1: $Ser_2 \sim Ser_3$, so implement $(2, 2)$-scheme on *SI* to spawn two random shares say, $A_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $B_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}$. Assign A_1 to Ser_1 and B_1 to both Ser_2 and Ser_3.
2) For Γ_2: Implement $(2, 2)$-scheme on *SI* to spawn two new random shares $A_2 = \begin{bmatrix} 0 & 0 \end{bmatrix}$ and $B_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Implement $(2, 2)$-scheme on B_2 to spawn shares $A_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and $B_3 = \begin{bmatrix} 0 & 1 \end{bmatrix}$.Then distribute A_2 to Ser_2, A_3 to Ser_3 and B_3 to Ser_4 respectively.
So Ser_1 holds share $sh_{(1,1)} = A_1$. Ser_2 holds shares $sh_{(2,1)} = B_1$ and $sh_{(2,2)} = A_2$. Ser_3 holds shares $sh_{(3,1)} = B_1$ and $sh_{(3,2)} = A_3$. Ser_4 holds share $sh_{(4,1)} = B_3$. The individual

random shares generated using Liu et al.'s step construction for XVCS are of same size of *SI*. Since the servers hold multiple shares, the average pixel expansion APE $= (1+2+2+1)/4$. Reconstructed image *RI* is exactly same as *SI* and can be generated jointly by

1) Servers Ser_1 and Ser_2 by computing: $RI = sh_{(1,1)} \oplus sh_{(2,1)}$.
2) Servers Ser_1 and Ser_3 by computing: $RI = sh_{(1,1)} \oplus sh_{(3,1)}$.
3) Servers Ser_2, Ser_3 and Ser_4 by computing: $RI = sh_{(2,2)} \oplus sh_{(3,2)} \oplus sh_{(4,1)}$.

2.4 VCS with Covers [6]

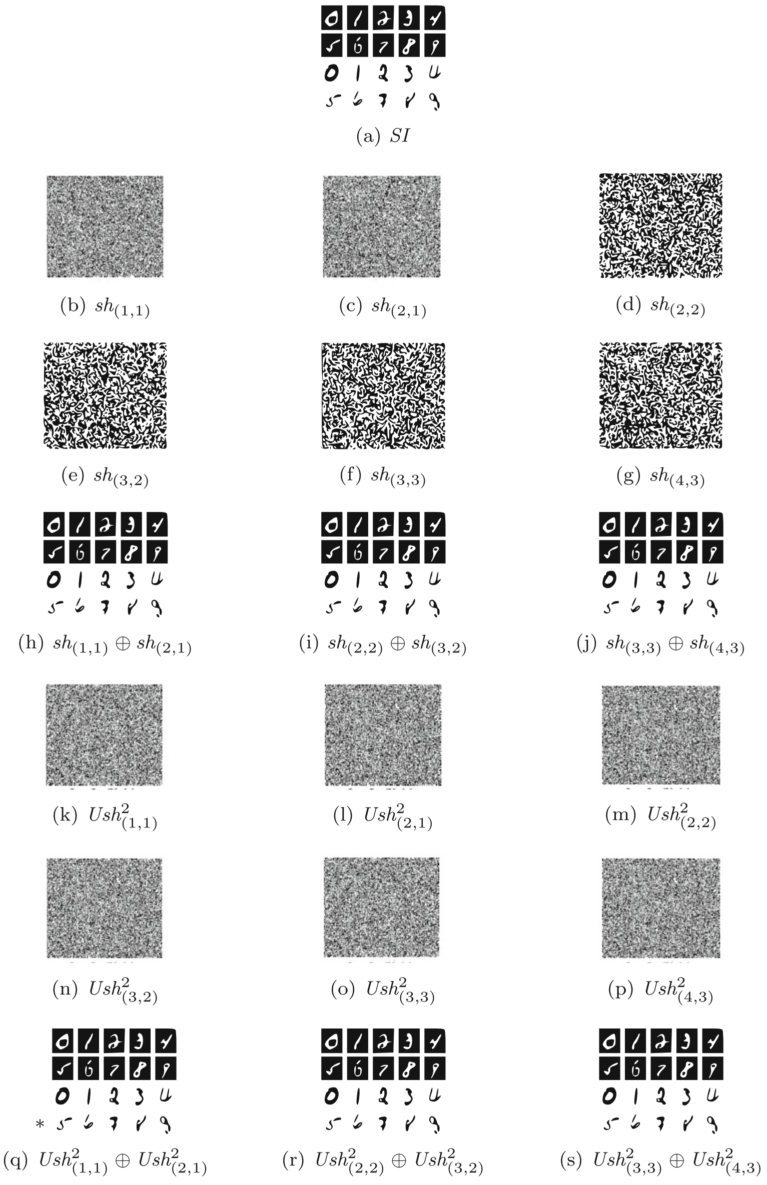
Meaningful shares are generally used to reduce the suspicion of(channel) attackers during the transmission of shares and facilitate share management. For VCS with covers, the share of the servers belonging to set *Ser* is meaningful, and is not random-looking as in conventional VCS. Let $C^{sp}_{b_1,..,b_n}$ for, b_1, .., $b_n \in \{0, 1\}$, be the collection of matrices to encode a pixel b_i, where $i = 1$ to n in the image Cov_i (meaningful images or cover images) associated to servers in set *Ser* in order to obtain a secret pixel *sp* when the transparencies associated to the servers in the set $A \in \Gamma_{Qual}$ are stacked together. Hence there will be a collection of 2^n pairs $(C^0_{b_1,..,b_n}, C^1_{b_1,..,b_n})$ for all possible combinations of white and black pixels in the n cover images. Let $T^0_{b_1,..,b_n} = \left[S^0 \,||\, H\right] \in C^0_{b_1,..,b_n}$ and $T^1_{b_1,..,b_n} = \left[S^1 \,||\, H\right] \in C^1_{b_1,..,b_n}$ where, S^0 (resp. S^1) are the basis matrices of a based perfect black VCS [5] and when "OR" ing the rows of H matrix corresponding to the servers in the Γ_{QM} (qualified set), an all one row vector will obtain. This implies that T^0 (resp. T^1) are basis matrices of a perfect black extended VCS for sharing 0 (resp. 1) pixel in *SI*. Example 2 illustrates this extended VCS construction.

Example 2 Let $Ser = \{Ser_1, Ser_2, Ser_3\}$ be the set of servers. Let the minimal qualified set is denoted by $\Gamma_{QM} = \{\{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}, \{Ser_2, Ser_3\}\}$. Let $SI = \begin{bmatrix}0 & 1\end{bmatrix}$. Let the cover images used are $Cov_1 = \begin{bmatrix}1 & 1\end{bmatrix}$, $Cov_2 = \begin{bmatrix}0 & 0\end{bmatrix}$ and $Cov_3 = \begin{bmatrix}1 & 0\end{bmatrix}$. Let $S^0 = \begin{bmatrix}1&1&0&1\\1&1&0&1\\1&1&0&1\end{bmatrix}$ and $S^1 = \begin{bmatrix}1&1&0&1\\1&1&1&0\\1&0&1&1\end{bmatrix}$. For sharing a 0 pixel and 1 pixel, the Boolean matrices used are $T^0_{101} = \left[\begin{array}{cccc||ccc}1&1&0&1&1&1&1\\1&1&0&1&0&1&1\\1&1&0&1&1&1&1\end{array}\right]$ and $T^1_{100} = \left[\begin{array}{cccc||ccc}1&1&0&1&1&1&1\\1&1&1&0&0&1&1\\1&0&1&1&1&0&1\end{array}\right]$ respectively. The pixel expansion and relative contrast of Ateniese et al. [6] scheme, for this access structure $(\Gamma_{QM}, \Gamma_{FM})$ is 7 and 1/7 respectively.

The following two sections contain the main results of the proactive visual cryptographic scheme. In Section 3, we describe the basic proactive schemes without cover images and later in Section 4, we will extend the basic scheme to include meaningful cover images.

3 Proactive VCS (PS) without covers

In this section, we present two efficient proactive XOR-based visual cryptographic schemes (XVCS) for any general access structure. The first proactive scheme (**PS-1**) follows a linear algebraic construction methodology (Fig. 3). The second scheme (**PS-2**) uses the step construction technique of Liu et al. [21] for generating the shares of secret pixel values. We



(a) SI

(b) $sh_{(1,1)}$ (c) $sh_{(2,1)}$ (d) $sh_{(2,2)}$

(e) $sh_{(3,2)}$ (f) $sh_{(3,3)}$ (g) $sh_{(4,3)}$

(h) $sh_{(1,1)} \oplus sh_{(2,1)}$ (i) $sh_{(2,2)} \oplus sh_{(3,2)}$ (j) $sh_{(3,3)} \oplus sh_{(4,3)}$

(k) $Ush^2_{(1,1)}$ (l) $Ush^2_{(2,1)}$ (m) $Ush^2_{(2,2)}$

(n) $Ush^2_{(3,2)}$ (o) $Ush^2_{(3,3)}$ (p) $Ush^2_{(4,3)}$

(q) $Ush^2_{(1,1)} \oplus Ush^2_{(2,1)}$ (r) $Ush^2_{(2,2)} \oplus Ush^2_{(3,2)}$ (s) $Ush^2_{(3,3)} \oplus Ush^2_{(4,3)}$

Fig. 3 Experimental results based on **PS-1** for access structure $\Gamma = \{\{Ser_1, Ser_2\}, \{Ser_2, Ser_3\}, \{Ser_3, Ser_4\}\}$: (a) Secret image, (b-g) shares of servers Ser_1, Ser_2, Ser_3, Ser_4 with out any renewals, (h-j) Reconstructed outputs without any renewals, (k-p) shares of servers after second renewal, (g-i) Reconstructed outputs at time $t = 2$

describe the renewal procedures and perform a comparative analysis of the two schemes. We observe that the renewal procedure does not affect the quality of the reconstructed image for both schemes.

3.1 Proactive linear algebraic XVCS (PS-1)

We begin with describing the basic visual cryptographic scheme for general access structure following linear algebra-based methodology on binary finite field Z_2. Dutta and Adhikari [9] proposed a linear algebraic construction of XVCS for any threshold access structures but their construction was through basis matrices which are unsuitable for incorporating proactivity into the system. The following construction resolves that problem and we will be able to add a renewal procedure to update the shares.

Let Γ =$(\Gamma_{QM}, \Gamma_{FM})$ be a (general) access structure on a set of n servers. The share generation algorithm share_Γ is as follows.
We associate a Boolean variable x_i to each server i for all $i = 1, 2, \ldots, n$. $B \in \Gamma_{QM}$ is a typical representative of a minimal qualified set in the access structure. We arrange the elements of Γ_{QM} in some order according to our choice say, e.g., lexicographic order, $B_1, B_2, \ldots, B_r$.

For each B_q where $1 \leq q \leq r$ consider the equations:
$f_{B_j} = 0$ and $f_{B_j} = 1$ which respectively denote the linear equations $\sum\limits_{k \in B_j} x_k = 0(mod\ 2)$ and $\sum\limits_{k \in B_j} x_k = 1(mod\ 2)$. Also, denote by $C_j = \mathcal{P} \setminus B_j$ i.e. those servers who are not in B_j.
Further, let $\boldsymbol{\mathcal{F}}_{\mathbf{C_i}} = \mathbf{0}$ denote the following system of linear equations:

$$x_{i_1} = 0,\ x_{i_2} = 0, \ldots,\ x_{i_{t_i}} = 0.$$

where $C_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{t_i}}\}$.

We consider the following systems of linear equations over the field Z_2:
For $1 \leq i \leq r$,

$$\left.\begin{aligned} f_{B_i} &= 0 \\ \boldsymbol{\mathcal{F}}_{\mathbf{C_i}} &= \mathbf{0} \end{aligned}\right\} \cdots (i) \qquad \text{and} \qquad \left.\begin{aligned} f_{B_i} &= 1 \\ \boldsymbol{\mathcal{F}}_{\mathbf{C_i}} &= \mathbf{0} \end{aligned}\right\} \cdots (i')$$

Let for any $1 \leq i \leq r$, S_i^0 denote the Boolean matrix whose columns are all possible solutions of the system (i). Also, let S_i^1 denote the Boolean matrix whose columns are all possible solutions of the system (i'). That is, if we choose the entries indexed by the servers in B_i from any column of S_i^0, then those entries satisfy the equation $\sum\limits_{k \in B_i} x_k = 0(mod\ 2)$. Same is the intuition for S_i^1. The above observations show that when a qualified set B_i of shares are "superposed" then they reconstruct the white/black pixel perfectly.

The share generation algorithm share_Γ is summarized in Fig. 4. Note that, the share generation algorithm assigns multiple shares to a server and in fact, for every minimal qualified set in the access structure, a share is assigned to a server. Example 3 shows the share generation procedure.

The reconstruction procedure is simple. When a qualified set $Q \in \Gamma_{QM}$ of servers submits shares for reconstruction then a minimal qualified set $B \subseteq Q$ of shares are chosen and the secret is reconstructed using the shares corresponding to that minimal qualified set.

A Fact from Linear Algebra. It is a well known fact that if we consider two systems of linear equations (written using the matrix notation) $Ax = 0$ and $Ax = b$ where $b \neq 0$, then all possible solutions of the second system can be obtained by adding (i.e., addition of

```
share$_{\Gamma,\mathcal{P}}(.)$:
  – Input: secret pixel $b \in \{0,1\}$ and access structure $\Gamma$ on server set $\mathcal{P}$.
    – For each server $j \in \mathcal{P}$, define a variable $x_j$.
    – For each minimal qualified set $B \in \Gamma$:
        • solve: $\sum_{k \in B} x_k = b \mod 2$ & $x_{k'} = 0$ for all $k' \notin B$
        • Construct the matrix $S_B^b$ of size $n \times 2^{k-1}$: the columns are solutions to the above equation.
        • randomly choose a column from $S_B^b$ and assign the $j^{th}$ entry as the share of server $j \in B$.
    – Output: Every server $j \in \mathcal{P}$ receives one bit as share information for each minimal qualified set $B \in \Gamma$ for which $j \in B$.
```

Fig. 4 share$_\Gamma(b)$: share generation algorithm for a secret pixel b

solution vectors) one particular solution of the second system to each solution of the first system.

Remark 1 The methodology followed to construct S_i^0 and S_i^1 (as described above) shows that each block S_i^1 can be obtained from S_i^0 by adding a particular solution of the system (i') to each column of S_i^0. Therefore, it is not so hard to see that for a forbidden set F of servers the restricted submatrix $S_i^0[F]$ (which contains only the rows corresponding to the indices in F) is equal to $S_i^1[F]$ (maybe the columns are permuted).

Example 3 Let $Ser = \{Ser_1, Ser_2, Ser_3, Ser_4\}$ and $SI = \begin{bmatrix} 0 & 1 \end{bmatrix}$ denote the set of servers and secret image respectively. Let the minimal qualified set is denoted by $\Gamma_{QM} = \{\{Ser_1, Ser_2\}, \{Ser_2, Ser_3\}, \{Ser_3, Ser_4\}\}$.

(1) In order to spawn shares for the secret pixel 0 solve the equations $x_1 \oplus x_2 = 0$; $x_1 \oplus x_3 = 0$; $x_3 \oplus x_4 = 0$ and construct the three matrices $S_1^0 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$; $S_2^0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$; $S_3^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$. To spawn the shares of the servers choose one column randomly and distribute the entries to the servers. Suppose we choose first column of S_1^0, second column of S_2^0 and second column of S_3^0 then the shares of Ser_1 are $\{0, 1, 0\}$. shares of Ser_2 are $\{0, 0, 0\}$, for Ser_3 are $\{0, 1, 0\}$ and for Ser_4 are $\{0, 0, 1\}$.

(2) In order to spawn shares for the secret pixel 0 solve the equations $x_1 \oplus x_2 = 1$; $x_1 \oplus x_3 = 1$; $x_3 \oplus x_4 = 1$ and construct the three matrices

$$S_1^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; S_2^1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}; S_3^1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

As before we can choose random columns to spawn shares for the servers.

We now have the following theorem.

Theorem 2 *For a given access structure* Γ *on a set of servers* $\mathcal{P}$*, the share generation algorithm* share$_{\Gamma,\mathcal{P}}$ *as described in* Fig. 4 *admits an* XVCS *satisfying the conditions of Definition 4.*

3.1.1 Share update process (PS-1)

Initial Setup & Input : Let SI, D, Ser_u for $1 \leq u \leq n$, y_u and $\Gamma = (\Gamma_{QM}, \Gamma_{FM})$ respectively denote the secret image, dealer, n servers, number of shares hold by u^{th} server Ser_u and a general access structure. Initially D runs $\mathsf{share}_\Gamma(SI)$ using share generation algorithm presented in Fig. 4 to spawn $sh_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$ and distributes them with Ser_u for $1 \leq u \leq n$ respectively.

Share Renewal process at time t **:** Suppose at time $t-1$ the shares (after $t-2$ renewals) of the servers are $Ush^{(t-1)}_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$. Each server Ser_u for $1 \leq u \leq n$ perform following steps:

1. Run $\mathsf{share}_\Gamma(0)$ to spawn $sh^t_{(u,k,j)}[0]$ for $1 \leq k \leq n$, $1 \leq j \leq y_u$
2. Stores $sh^t_{(u,u,j)}[0]$ and sends (through a secure channel) $sh^t_{(u,k,j)}[0]$ to server Ser_k for all $1 \leq k(\neq u) \leq n,\ 1 \leq j \leq y_u$.
3. Compute $\mathtt{othS}^t_{(u,j)}[0] = \oplus_{k \neq u} sh^t_{(k,u,j)}[0]$ for all $1 \leq k(\neq u) \leq n$, $1 \leq j \leq y_u$.
4. Outputs $Ush^{(t)}_{(u,j)} \longleftarrow Ush^{(t-1)}_{(u,j)} \oplus sh^t_{(u,u,j)}[0] \oplus \mathtt{othS}^t_{(u,j)}[0]$.
5. Lastly, each server stores this updated value $Ush^{(t)}_{(u,j)}$ into it and deletes the old share.

3.1.2 Secret reconstruction phase

The reconstruction procedure remains the same as the underlying XVCS which has been used as the basic building block - viz., any minimal set B of servers can submit their shares corresponding to B and simply output the "exclusive-or" (xor) of the shares. It is not hard to see that since at each share renewal step, 0 is shared by all the servers therefore, by the linearity of the reconstruction process the correct secret is output even after t many renewal procedures.

Table 3 shows the pixel expansion and relative contrast of **PS-1** for some of the access structures. Experimental results of **PS-1** is provided in Fig. 3 for access structure

Table 3 Average Pixel expansion (APE) and relative contrast for some access structures on at most four servers: APE is computed by averaging the total pixels stored by all servers for each secret pixel divided by the number of servers; APE in PS-1 is no better than PS-2

Γ_{QM}	PS-1(PE,α)	PS-2(APE, α)
$\{Ser_1, Ser_2\}$	(1,1)	(1, 1)
$\{Ser_1, Ser_2, Ser_3\}$	(1,1)	(1, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_2, Ser_3\}$,$\{Ser_3, Ser_4\}$	(3,1)	(1.25, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_1, Ser_3\}$,$\{Ser_1, Ser_4\}$	(3,1)	(1, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_1, Ser_4\}$,$\{Ser_2, Ser_3\}$,$\{Ser_3, Ser_4\}$	(4,1)	(1, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_2, Ser_3\}$,$\{Ser_2, Ser_4\}$,$\{Ser_3, Ser_4\}$	(4,1)	(1.50, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_4\}$	(2,1)	(1, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_4\}$,$\{Ser_3, Ser_4\}$	(3,1)	(1.50, 1)
$\{Ser_1, Ser_3, Ser_4\}$,$\{Ser_1, Ser_2\}$,$\{Ser_2, Ser_3\}$,$\{Ser_2, Ser_4\}$	(4,1)	(1.75, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_2, Ser_4\}$	(2,1)	(1,1)
$\{Ser_1, Ser_2, Ser_4\}$,$\{Ser_1, Ser_3, Ser_4\}$,$\{Ser_2, Ser_3\}$	(3,1)	(1.50, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_2, Ser_4\}$,$\{Ser_1, Ser_3, Ser_4\}$	(3,1)	(1.50,1)
$\{Ser_1, Ser_2, Ser_3, Ser_4\}$	(1,1)	(1,1)

$\Gamma = \{\{Ser_1, Ser_2\}, \{Ser_2, Ser_3\}, \{Ser_3, Ser_4\}\}$. The reconstructed output is same as secret image when the qualified subsets of servers (eg: $\{Ser_1, Ser_2\}$) combine their corresponding shares during with out renewals(eg: $sh_{(1,1)} \oplus sh_{(2,1)}$) and with renewals (eg: $Ush^2_{(1,1)} \oplus Ush^2_{(2,1)}$).

3.2 Proactive step construction based XVCS (PS-2)

We now describe our second scheme which is more efficient in terms of pixel expansion (share size) and the comparison with the previous construction can be found in Table 3.

3.2.1 Share update process (PS-2)

Initial Setup & Input : Let *SI*, *D*, Ser_u for $1 \leq u \leq n$, y_u and Γ =(Γ_{QM}, Γ_{FM}) respectively denote the secret image, dealer, *n* servers, number of shares hold by u^{th} server Ser_u and a general access structure. Initially *D* runs share generation algorithm based on Liu et al. [21] discussed in Section 2.3 to spawn $sh_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$ and distributes them with Ser_u for $1 \leq u \leq n$ respectively.

Share Renewal process at time t **:** Similar to given in Section 3.1.1. The only modification is that the share (of 0) generated by each server during the renewal process follows the step construction of Liu et al. for the given access structure.

3.2.2 Secret reconstruction phase

Same as secret reconstruction algorithm based on Liu et al. [21] discussed in Section 2.3.

Tables 3 and 4 shows the pixel expansion and relative contrast of **PS-2** for some of the access structures. Detailed explanation of **PS-2** is given in Example 4. Experimental results of **PS-2** is provided in Fig. 5 for $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$. The reconstructed output is same as secret image when the qualified subsets of servers (eg: $\{Ser_1, Ser_2, Ser_3\}$) combine their corresponding shares without renewals (eg: $sh_{(1,1)} \oplus sh_{(2,1)} \oplus sh_{(3,1)}$) and with renewals (eg: $Ush^2_{(1,1)} \oplus Ush^2_{(2,1)} \oplus Ush^2_{(3,1)}$).

Example 4 Consider same server set *Ser*, secret image *SI* and access structure Γ_{QM} as given in Example 1. Then Ser_1 holds share $Ush^0_{(1,1)} = sh_{(1,1)}$, Ser_2 holds shares $Ush^0_{(2,1)} = sh_{(2,1)}$ and $Ush^0_{(2,2)} = sh_{(2,2)}$, Ser_3 holds shares $Ush^0_{(3,1)} = sh_{(3,1)}$ and $Ush^0_{(3,2)} = sh_{(3,2)}$, Ser_4 holds share $Ush^0_{(4,1)} = sh_{(4,1)}$. Let $Z = \begin{bmatrix} 0 & 0 \end{bmatrix}$. Then as per step construction by Liu et al. developed on XOR operation it is possible to divide Γ_{QM} into two parts $\Gamma_1 = \{\{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}\}$ and $\Gamma_2 = \{Ser_2, Ser_3, Ser_4\}$. The tasks done by each server Ser_u for $1 \leq u \leq n$ are listed below.

Ser_1 will do the following.

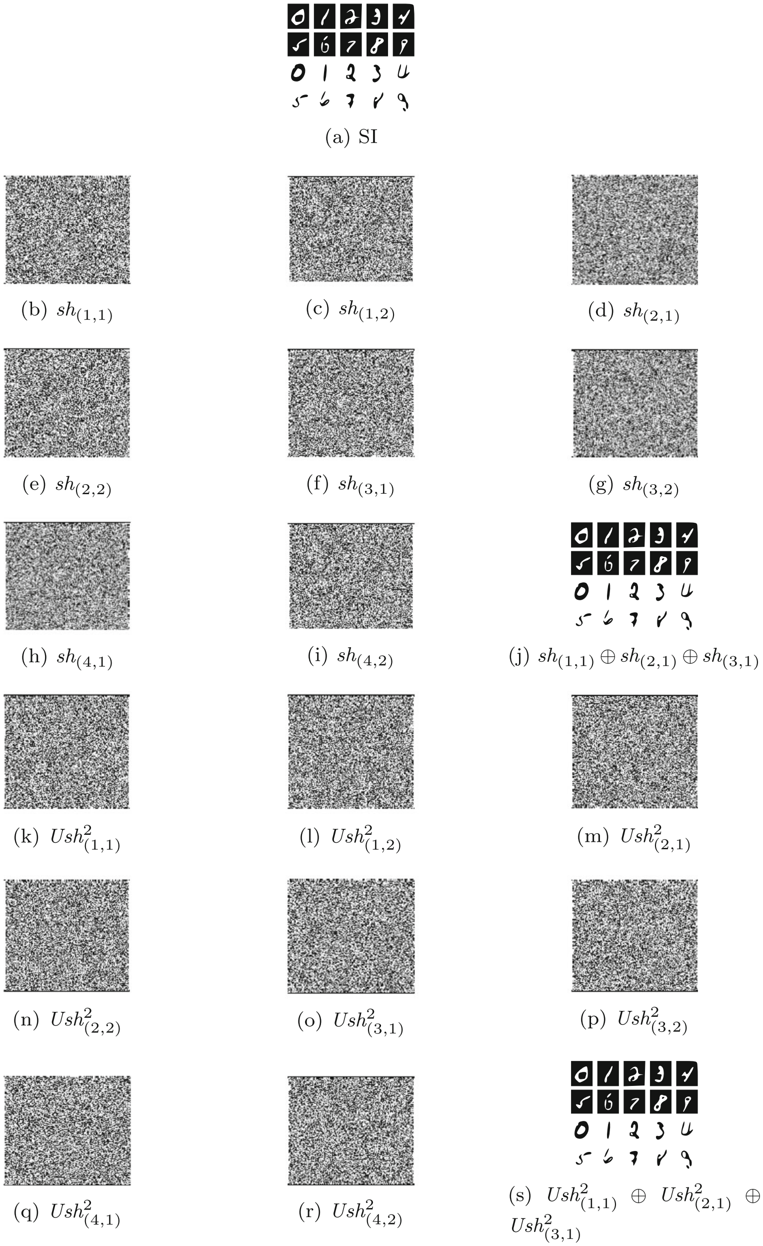
1) In the case of Γ_1, $Ser_2 \sim Ser_3$, so Implement (2, 2)-scheme on *Z* to spawn two shares $A_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and $B_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Then store $sh^1_{(1,1,1)}[0] = A_1$, send $sh^1_{(1,2,1)}[0] = B_1$ to Ser_2 and $sh^1_{(1,3,1)}[0] = B_1$ to Ser_3 respectively.
2) In the case of Γ_2, again Implement (2, 2)-scheme on *Z* to spawn two new shares $A_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $B_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}$. Implement (2, 2)-scheme on B_2 to spawn shares $A_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and $B_3 = \begin{bmatrix} 1 & 0 \end{bmatrix}$.Then distribute $sh^1_{(1,2,2)}[0] = A_2$ to Ser_2, $sh^1_{(1,3,2)}[0] = A_3$ to Ser_3 and $sh^1_{(1,4,1)}[0] = B_3$ to Ser_4 respectively.

Table 4 Pixel expansion and relative contrast for (k, n) access structures: PS-2(SC) is for single cover & PS-2(DC) stands for different covers

(k, n)	PS-2(APE, α)	PS-2(SC)(APE, α)	PS-2(DC)(APE, α)
(2, 3)	(1.60, 1)	(10.8, 1)	(12.6, 1)
(2, 4)	(2.25, 1)	(12.75,1)	(16.5,1)
(3, 4)	(2, 1)	(12, 1)	(15, 1)
(2, 5)	(2.80, 1)	(14.40, 1)	(19.8, 1)
(3, 5)	(3.6, 1)	(16.8, 1)	(24.6, 1)
(4, 5)	(2.6, 1)	(13.8, 1)	(18.6, 1)
(2, 6)	(3.33, 1)	(16, 1)	(23, 1)
(3, 6)	(5.5, 1)	(22.5, 1)	(36, 1)
(4, 6)	(5.33, 1)	(21.9, 1)	(34.8, 1)
(5, 6)	(2.6, 1)	(13.98, 1)	(18.98, 1)
(2, 7)	(3.85, 1)	(17.57, 1)	(26.14, 1)
(3, 7)	(7.71, 1)	(24, 1)	(39, 1)
(4, 7)	(9.42, 1)	(34.26, 1)	(59.52, 1)
(5, 7)	(6.85, 1)	(26.55, 1)	(44.1, 1)
(6, 7)	(3.28, 1)	(15.84, 1)	(22.68, 1)
(2, 8)	(4.37, 1)	(19.12, 1)	(29.24, 1)
(3, 8)	(10.25, 1)	(36.75, 1)	(64.5, 1)
(4, 8)	(15.12, 1)	(51.37, 1)	(93.74, 1)
(5, 8)	(14.25, 1)	(48.75, 1)	(88.5, 1)
(7, 8)	(3.25, 1)	(15.75, 1)	(22.5, 1)
(2, 9)	(4.88, 1)	(20.66, 1)	(32.32, 1)
(3, 9)	(13.11, 1)	(45.33, 1)	(81.66, 1)
(4, 9)	(22.66, 1)	(75.98, 1)	(142.96, 1)
(5, 9)	(26.11, 1)	(84.33, 1)	(159.66, 1)
(8, 9)	(3.88, 1)	(17.4, 1)	(25.8, 1)
(2, 10)	(5.40, 1)	(22.20, 1)	(35.4, 1)
(3,10)	(16.3, 1)	(54.9, 1)	(100.8, 1)
(4,10)	(32.3, 1)	(102.9, 1)	(196.8, 1)
(5,10)	(43.9, 1)	(137.7, 1)	(266.4, 1)
(9,10)	(3.80, 1)	(17.4, 1)	(25.8, 1)

Ser_2 will do the following.

1) In the case of Γ_1, $Ser_2 \sim Ser_3$, so Implement $(2, 2)$-scheme on Z to spawn two shares $A_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and $B_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$. Then send $sh^1_{(2,1,1)}[0] = A_1$ to Ser_1, store $sh^1_{(2,2,1)}[0] = B_1$ and send $sh^1_{(2,3,1)}[0] = B_1$ to Ser_3 respectively.
2) In the case of Γ_2, again Implement $(2, 2)$-scheme on Z to spawn two new shares $A_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}$ and $B_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Implement $(2, 2)$-scheme on B_2 to spawn shares $A_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and $B_3 = \begin{bmatrix} 0 & 1 \end{bmatrix}$.Then store $sh^1_{(2,2,2)}[0] = A_2$, send $sh^1_{(2,3,2)}[0] = A_3$ to Ser_3 and $sh^1_{(2,4,1)}[0] = B_3$ to Ser_4 respectively.



(a) SI

(b) $sh_{(1,1)}$

(c) $sh_{(1,2)}$

(d) $sh_{(2,1)}$

(e) $sh_{(2,2)}$

(f) $sh_{(3,1)}$

(g) $sh_{(3,2)}$

(h) $sh_{(4,1)}$

(i) $sh_{(4,2)}$

(j) $sh_{(1,1)} \oplus sh_{(2,1)} \oplus sh_{(3,1)}$

(k) $Ush^2_{(1,1)}$

(l) $Ush^2_{(1,2)}$

(m) $Ush^2_{(2,1)}$

(n) $Ush^2_{(2,2)}$

(o) $Ush^2_{(3,1)}$

(p) $Ush^2_{(3,2)}$

(q) $Ush^2_{(4,1)}$

(r) $Ush^2_{(4,2)}$

(s) $Ush^2_{(1,1)} \oplus Ush^2_{(2,1)} \oplus Ush^2_{(3,1)}$

Fig. 5 Experimental results based on **PS-2** for access structure $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$: (a) Secret image, (b-i) shares of servers Ser_1, Ser_2, Ser_3 and Ser_4 without any renewals, (j) Reconstructed output without any renewals, (k-r) shares of servers after second renewal, (s) Reconstructed output at time $t = 2$

Similarly Ser_3 and Ser_4 will do the same process as mentioned above.

Now each server Ser_u for $1 \leq u \leq n$ can do the share update process as shown below.

Ser_1 will do the following.

1) Compute $\mathtt{othS}^1_{(1,1)}[0] = sh^1_{(2,1,1)}[0] \oplus sh^1_{(3,1,1)}[0] \oplus sh^1_{(4,1,1)}[0]$.
2) Outputs $Ush^{(1)}_{(1,1)} \longleftarrow Ush^0_{(1,1)} \oplus sh^1_{(1,1,1)}[0] \oplus \mathtt{othS}^1_{(1,1)}[0]$.

Ser_2 will do the following.

1) Compute $\mathtt{othS}^1_{(2,1)}[0] = sh^1_{(1,2,1)}[0] \oplus sh^1_{(3,2,1)}[0] \oplus sh^1_{(4,2,1)}[0]$ and $\mathtt{othS}^1_{(2,2)}[0] = sh^1_{(1,2,2)}[0] \oplus sh^1_{(3,2,2)}[0] \oplus sh^1_{(4,2,2)}[0]$
2) Outputs $Ush^{(1)}_{(2,1)} \longleftarrow Ush^0_{(2,1)} \oplus sh^1_{(2,2,1)}[0] \oplus \mathtt{othS}^1_{(2,1)}[0]$ and $Ush^{(1)}_{(2,2)} \longleftarrow Ush^0_{(2,2)} \oplus sh^1_{(2,2,2)}[0] \oplus \mathtt{othS}^1_{(2,2)}[0]$.

Similarly Ser_3 and Ser_4 will compute and output shares.

Then, reconstructed image *RI* is same as *SI* and is generated jointly by

1) Servers Ser_1 and Ser_2 by doing: $RI = Ush^1_{(1,1)} \oplus Ush^1_{(2,1)}$.
2) Servers Ser_1 and Ser_3 by doing: $RI = Ush^1_{(1,1)} \oplus Ush^1_{(3,1)}$.
3) Servers Ser_2, Ser_3 and Ser_4 by doing: $RI = Ush^1_{(2,2)} \oplus Ush^1_{(3,2)} \oplus Ush^1_{(4,1)}$.

The average pixel expansion of the scheme is APE $= \frac{1+(1+1)+(1+1)+1}{4} = 1.5$.

4 Proactive VCS with meaningful cover images

VCS with cover shares was proposed by Ateniese et al. [6] in 2001. Praveen et al. [17] in 2019 proposed a VCS with with cover shares, by extending step construction by Liu et al. [21]. In this section, we present two efficient proactive XOR-based visual cryptographic schemes (XVCS) for any general access structure with meaningful shares. In the first proactive scheme with meaningful shares (**PS-2(SC)**) we use the same cover images for construction, while in the second scheme (**PS-2(DC)**) different cover images are used for constructing shares on top of our **PS-2** scheme. We describe the renewal procedures and perform a comparative analysis of the two schemes. We observe that the renewal procedure does not affect the quality of the reconstructed image for both schemes. The *tuning* procedure included in our modified reconstruction algorithm guarantees the lossless retrieval of the secret image. In Figs. 6 and 9 we give an outline of the procedures.

4.1 Proactive XVCS with same covers for general access structure (PS-2(SC))

Let D, Ser_u for $1 \leq u \leq n$, y_u and Γ =$(\Gamma_{QM}, \Gamma_{FM})$ respectively denote the dealer, n servers, number of shares hold by server Ser_u for $1 \leq u \leq n$ and a general access structure.

4.1.1 Permutation share and Expanded cover share generation

Corresponding to each pixel in the cover image $\{Cov(g,h) : 1 \leq g \leq p, 1 \leq h \leq q\}$, the dealer D will do the following.

1. Implement a random column permutation to the matrix $De = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$ and append De with the share M.
2. The same cover pixel $Cov(g, h)$ is appended three times with the share $Ecov$.

Now dealer D will send M and $Ecov$ of size $p \times 3q$ to each server Ser_u for $1 \leq u \leq n$. The detailed procedure for generating M and $Ecov$ is shown below.

```
Permutation Share and Expanded cover generation algorithms

Input :
 1. A cover image $\{Cov(g, h) : 1 \leq g \leq p, 1 \leq h \leq q\}$

Algorithm :
For $g = 1$ to $p$
    For $h = 1$ to $q$
        $De = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$
        Implement a random column permutation to the $De$ matrix which is of size $1 \times 3$.
            For $c = 1$ to $3$
                $M_{(g,h,c)} = De(c)$
                $Ecov_{(g,h,c)} = Cov(g, h)$
            End
    End
End

Output :
The permutation share $M$ and Extended cover share $Ecov$ which is of size $p \times 3q$ are given to all servers.
```

4.1.2 Meaningful share generation process

In order to spawn meaningful shares $Csh_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$ from random looking shares $sh_{(u,j)}$ corresponding to each server Ser_u, dealer D will make use of the same Permutation share M. Further, the same Permutation share M is used by each server Ser_u to update their shares during a particular time period. Corresponding to each server Ser_u, if the value of $M_{(g,h,t)} == 0$: assign the meaningful share pixel $Csh_{(u,j)}(g, h)$ as random looking share pixel $sh_{(u,j)}(g, h)$, otherwise append the meaningful share pixel $Csh_{(u,j)}(g, h)$ as the cover pixel $Cov(g, h)$, for $1 \leq g \leq p, 1 \leq h \leq q, 1 \leq c \leq 3$. The detailed procedure for generating meaningful shares $Csh_{(u,j)}$ is given below.

Meaningful share generation algorithm.

Input :

1. A permutation share M of size $p \times 3q$
2. Random looking shares $sh_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$
3. A Cover image $\{Cov(g,h) : 1 \leq g \leq p, 1 \leq h \leq q\}$

Algorithm :

```
For $g = 1$ to $p$
    For $h = 1$ to $q$
        For $j = 1$ to $y_u$
            For $u = 1$ to $n$
                For $c = 1$ to 3
                    if $M_{(g,h,c)} == 0$
                        $Csh_{(u,j)}(g,h,c) = sh_{(u,j)}(g,h)$
                    else
                        $Csh_{(u,j)}(g,h,c) = Cov(g,h)$
                End
            End
        End
    End
End
```

Output :

Set of meaningful shares, $\{Csh_{(u,j)} : 1 \leq u \leq n, 1 \leq j \leq y_u\}$ which are of size $p \times 3q$. The y_u meaningful shares, $Csh_{(u,j)}$ are given to u^{th} server Ser_u.

4.1.3 Proactive share generation and update process

1. Initially dealer D will do share generation process using **PS-2** by giving secret image SI input to generate random shares $sh_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$.
2. Then D runs the Permutation share and Expanded cover share generation phase.
3. Then run Meaningful share generation process which add covers to random shares $sh_{(u,j)}$ and obtain meaningful shares $Csh_{(u,j)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$.

Share Renewal process at time t (**PS-2(SC)**) :

Suppose at time $t-1$ the shares (after $t-2$ renewals) of the servers are $UCsh_{(u,j)}^{(t-1)}$ for $1 \leq u \leq n$, $1 \leq j \leq y_u$. Each server Ser_u for $1 \leq u \leq n$ performs the following steps:

1. Run share generation process using XOR based step construction of Liu et al. [21] scheme with input image as all zero image.
2. Then run Meaningful share generation process which add covers to random shares $sh_{(u,k,j)}^{t}[0]$ and obtain meaningful shares $Csh_{(u,k,j)}^{t}[0]$ for $1 \leq k \leq n$, $1 \leq j \leq y_u$.
3. Stores $Csh_{(u,u,j)}^{t}[0]$ and sends (through a secure channel) $Csh_{(u,k,j)}^{t}[0]$ to server Ser_k for all $1 \leq k(\neq u) \leq n$.

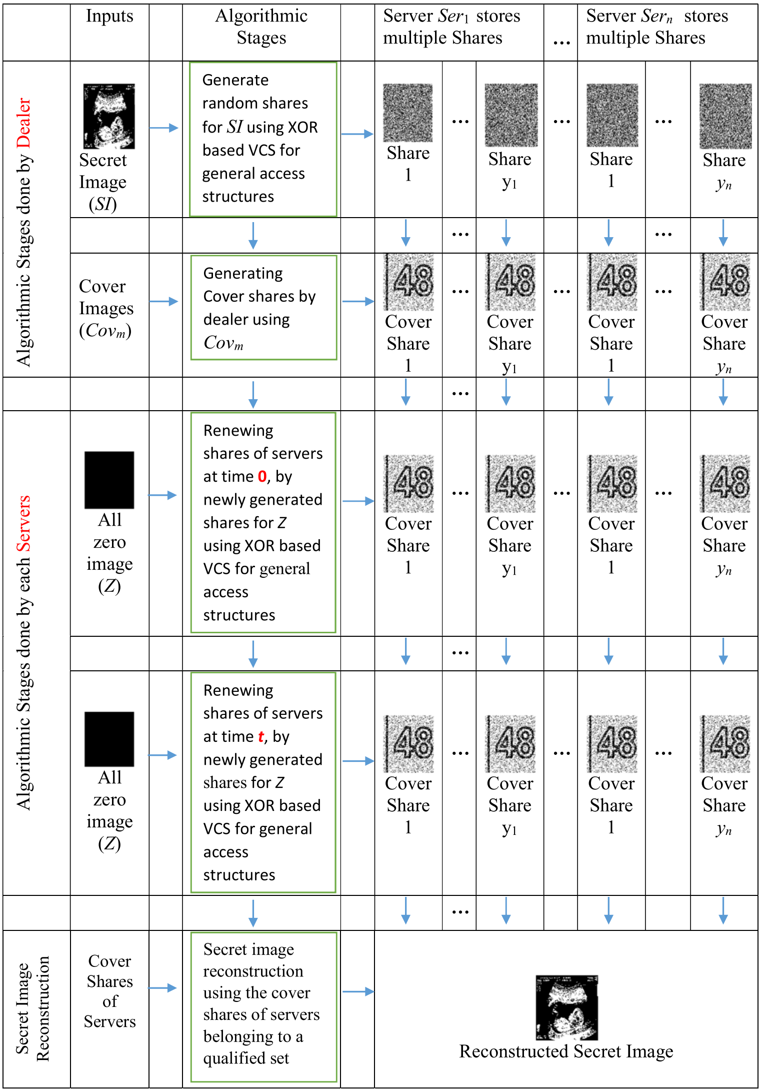

Inputs
Algorithmic Stages
Server Ser_1 stores multiple Shares
...
Server Ser_n stores multiple Shares
Algorithmic Stages done by Dealer
Secret Image (SI)
Generate random shares for SI using XOR based VCS for general access structures
Share 1
Share y_1
Share 1
Share y_n
Cover Images (Cov_m)
Generating Cover shares by dealer using Cov_m
Cover Share 1
Cover Share y_1
Cover Share 1
Cover Share y_n
Algorithmic Stages done by each Servers
All zero image (Z)
Renewing shares of servers at time 0, by newly generated shares for Z using XOR based VCS for general access structures
Cover Share 1
Cover Share y_1
Cover Share 1
Cover Share y_n
All zero image (Z)
Renewing shares of servers at time t, by newly generated shares for Z using XOR based VCS for general access structures
Cover Share 1
Cover Share y_1
Cover Share 1
Cover Share y_n
Secret Image Reconstruction
Cover Shares of Servers
Secret image reconstruction using the cover shares of servers belonging to a qualified set
Reconstructed Secret Image

Fig. 6 Algorithmic stages of our construction (**PS-2(SC)**): SC stands for same cover

4. If qualified set $B \in \Gamma_{QM}$ contains odd number of servers calculate $othS^t_{(u,j)}[0] = \oplus_{k \neq u} Csh^t_{(k,u,j)}[0]$, otherwise calculate $othS^t_{(u,j)}[0] = \oplus_{k \neq u} Csh^t_{(k,u,j)}[0] \oplus (M \odot Ecov)$. During the update process, in order to circumvent the cancelling effect of cover information from shares due to "XOR"($\oplus$) operation, $(M \odot Ecov)$ is used to make the shares meaningful. Here $(\odot)$ denotes "AND" operation.
5. Compute $UCsh^{(t)}_{(u,j)} \longleftarrow UCsh^{(t-1)}_{(u,j)} \oplus Csh^t_{(u,u,j)}[0] \oplus othS^t_{(u,j)}[0]$.

4.1.4 Secret reconstruction phase

1. In the case of qualified set $B \in \Gamma_{QM}$ which contains even number of servers follow the same scheme given in Section 2.3 to reconstruct the original reconstructed secret *RI*.
2. But in the case of qualified set $B \in \Gamma_{QM}$ which contains odd number of servers, first follow the same scheme given in Section 2.3 to spawn meaningful reconstructed secret *MRI*. For generating the original reconstructed secret *RI* do the following process: *RI* = *MRI* $\oplus$ $(M \odot Ecov)$.During the reconstruction process, in order to remove the cover information from *MRI*, $(M \odot Ecov)$ is used. Here $(\odot)$ denotes “AND” operation.
3. *Tuning Process*: This process will convert *RI* to an image (*PRI*) with same as that of *SI*. The detailed procedure is given below.

```
Tuning Process

Input :
 1. Reconstructed Image $RI$ of size $p \times 3q$

Algorithm :
For $g = 1$ to $p$
    For $h = 1$ to $q$
        For $c = 1$ to 3
            $Temp(c) = RI(g, h, c)$
        End
        if Hamming weight($Temp$)== 0
            $PRI(g, h) = 0$
        else
            $PRI(g, h) = 1$
    End
End

Output :
The perfectly reconstructed image $PRI$ which is the same as $SI$
```

4.1.5 Analysis on the pixel expansion & contrast

Tables 4 and 5 shows the APE and relative contrast of **PS-2(SC)** for some of the access structures. In this section we extended the scheme **PS-2** provided in Section 3.2 by adding covers to the random shares to generate meaningful shares. It is evident from the scheme that the pixel expansion of each meaningful shares *CSh* corresponding to each servers, permutation share *M* and the expanded cover share *Ecov* is 3. The dealer also is sharing both *M* and *Ecov* to all servers. So APE of this scheme **PS-2(SC)**= 3×APE(**PS-2**) + 3 + 3. Here the ratio of cover pixel and secret share pixel embedded within a meaningful share block of size 1 × 3 is 0.66 and 0.33 respectively after every renewal process and is given in Table 7. Detailed explanation of **PS-2(SC)** is given in Example 5 and Fig. 6 as a flowchart. Experimental results of **PS-2(SC)** is provided in Fig. 7 for access structure $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$. The reconstructed output is same as secret image when the qualified subsets of servers

Table 5 Average Pixel expansion (APE) and relative contrast for some access structures on at most four servers: APE using **PS-2(SC)** is better than using **PS-2(DC)**

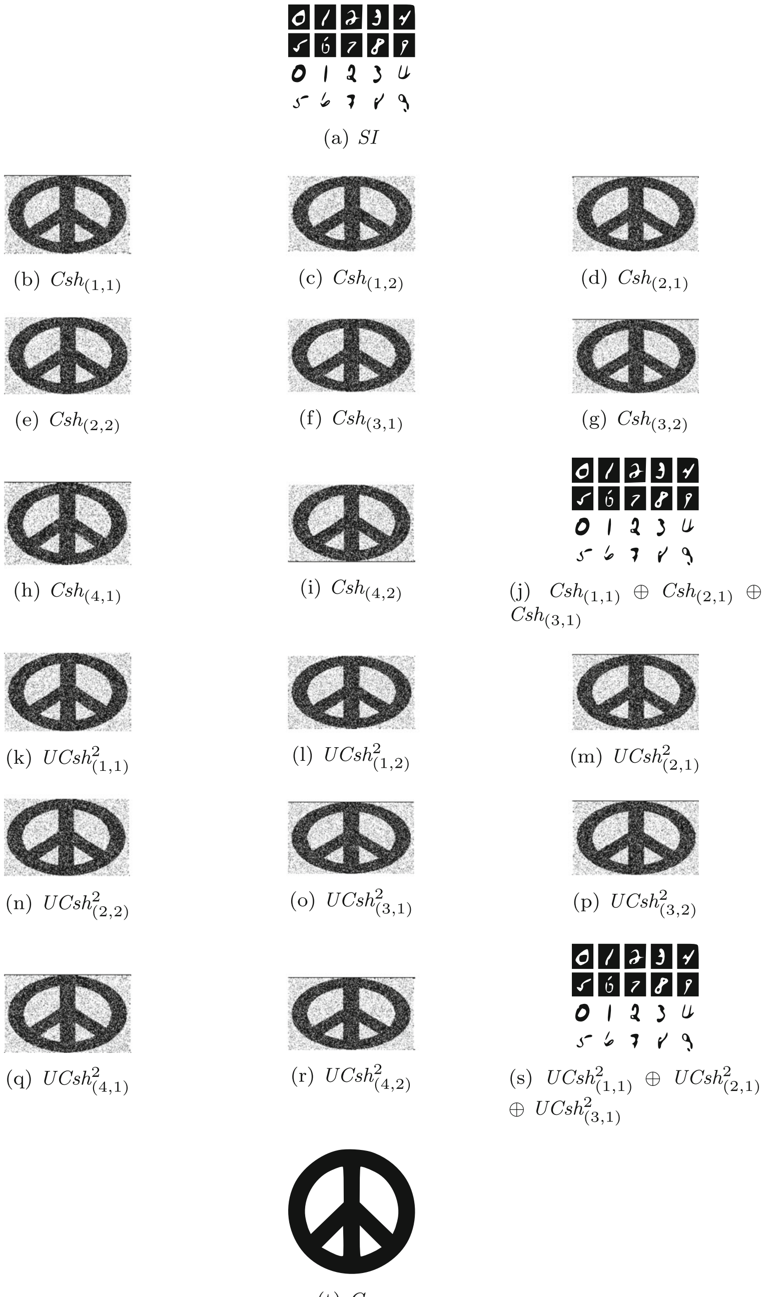
Γ_{QM}	PS-2(SC) (APE, α)	PS-2(DC) (APE, α)
$\{Ser_1, Ser_2\}$	(9, 1)	(9, 1)
$\{Ser_1, Ser_2, Ser_3\}$	(9, 1)	(9, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_2, Ser_3\}$,$\{Ser_3, Ser_4\}$	(9.75, 1)	(10.5, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_1, Ser_3\}$,$\{Ser_1, Ser_4\}$	(9, 1)	(9, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_1, Ser_4\}$,$\{Ser_2, Ser_3\}$,$\{Ser_3, Ser_4\}$	(9, 1)	(9, 1)
$\{Ser_1, Ser_2\}$,$\{Ser_2, Ser_3\}$,$\{Ser_2, Ser_4\}$,$\{Ser_3, Ser_4\}$	(10.50, 1)	(12, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_4\}$	(9, 1)	(9, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_4\}$,$\{Ser_3, Ser_4\}$	(10.50, 1)	(12, 1)
$\{Ser_1, Ser_3, Ser_4\}$,$\{Ser_1, Ser_2\}$,$\{Ser_2, Ser_3\}$,$\{Ser_2, Ser_4\}$	(11.25, 1)	(13.5, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_2, Ser_4\}$	(9, 1)	(9, 1)
$\{Ser_1, Ser_2, Ser_4\}$,$\{Ser_1, Ser_3, Ser_4\}$,$\{Ser_2, Ser_3\}$	(10.50, 1)	(12, 1)
$\{Ser_1, Ser_2, Ser_3\}$,$\{Ser_1, Ser_2, Ser_4\}$,$\{Ser_1, Ser_3, Ser_4\}$	(10.50, 1)	(12, 1)
$\{Ser_1, Ser_2, Ser_3, Ser_4\}$	(9, 1)	(9, 1)

(eg: $\{Ser_1, Ser_2, Ser_3\}$) combine their corresponding shares without renewals(eg: $Csh_{(1,1)} \oplus Csh_{(2,1)} \oplus Csh_{(3,1)}$) and with renewals (eg: $UCsh^2_{(1,1)} \oplus UCsh^2_{(2,1)} \oplus UCsh^2_{(3,1)}$). Tables 4 and 5 shows the comparison of **PS-2(SC)** with other schemes.

Example 5 Consider same *Ser*, *SI* and Γ_{QM} as given in Example 1. Let the cover images used is $Cov = \begin{bmatrix}0 & 1\end{bmatrix}$, $ECov = \begin{bmatrix}000 & 111\end{bmatrix}$ and $M = \begin{bmatrix}101 & 101\end{bmatrix}$. Then after adding covers the meaningful shares are $Csh_{(1,1)} = \begin{bmatrix}000 & 111\end{bmatrix}$, $Csh_{(2,1)} = \begin{bmatrix}010 & 111\end{bmatrix}$, $Csh_{(2,2)} = \begin{bmatrix}000 & 101\end{bmatrix}$, $Csh_{(3,1)} = \begin{bmatrix}010 & 111\end{bmatrix}$, $Csh_{(3,2)} = \begin{bmatrix}010 & 111\end{bmatrix}$, $Csh_{(4,1)} = \begin{bmatrix}000 & 111\end{bmatrix}$. Then Ser_1 holds share $UCsh^0_{(1,1)} = Csh_{(1,1)}$, Ser_2 holds shares $UCsh^0_{(2,1)} = Csh_{(2,1)}$ and $UCsh^0_{(2,2)} = Csh_{(2,2)}$, Ser_3 holds shares $UCsh^0_{(3,1)} = Csh_{(3,1)}$ and $UCsh^0_{(3,2)} = Csh_{(3,2)}$, Ser_4 holds share $UCsh^0_{(4,1)} = Csh_{(4,1)}$. Let $Z = \begin{bmatrix}0 & 0\end{bmatrix}$. Then as per step construction by Liu et al. developed on XOR operation it is possible to divide Γ_{QM} into two parts $\Gamma_1 = \{\{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}\}$ and $\Gamma_2 = \{Ser_2, Ser_3, Ser_4\}$. The tasks done by each server Ser_u for $1 \le u \le n$ are listed below by using the same random shares generated in Example 4.

Ser_1 will do the following.

1) In the case of Γ_1, after adding cover *Cov* to random shares it becomes, $A_1 = \begin{bmatrix}010 & 101\end{bmatrix}$ and $B_1 = \begin{bmatrix}010 & 101\end{bmatrix}$. Then store $Csh^1_{(1,1,1)}[0] = A_1$, send $Csh^1_{(1,2,1)}[0] = B_1$ to Ser_2 and $Csh^1_{(1,3,1)}[0] = B_1$ to Ser_3 respectively.
2) In the case of Γ_2, after adding cover *Cov* to random shares it becomes, $A_2 = \begin{bmatrix}000 & 111\end{bmatrix}$, $A_3 = \begin{bmatrix}010 & 111\end{bmatrix}$ and $B_3 = \begin{bmatrix}010 & 101\end{bmatrix}$.Then send $Csh^1_{(1,2,2)}[0] = A_2$ to Ser_2, $Csh^1_{(1,3,2)}[0] = A_3$ to Ser_3 and $Csh^1_{(1,4,1)}[0] = B_3$ to Ser_4 respectively.



(a) SI

(b) $Csh_{(1,1)}$ (c) $Csh_{(1,2)}$ (d) $Csh_{(2,1)}$

(e) $Csh_{(2,2)}$ (f) $Csh_{(3,1)}$ (g) $Csh_{(3,2)}$

(h) $Csh_{(4,1)}$ (i) $Csh_{(4,2)}$ (j) $Csh_{(1,1)} \oplus Csh_{(2,1)} \oplus Csh_{(3,1)}$

(k) $UCsh^2_{(1,1)}$ (l) $UCsh^2_{(1,2)}$ (m) $UCsh^2_{(2,1)}$

(n) $UCsh^2_{(2,2)}$ (o) $UCsh^2_{(3,1)}$ (p) $UCsh^2_{(3,2)}$

(q) $UCsh^2_{(4,1)}$ (r) $UCsh^2_{(4,2)}$ (s) $UCsh^2_{(1,1)} \oplus UCsh^2_{(2,1)} \oplus UCsh^2_{(3,1)}$

(t) Cov

Fig. 7 Experimental results based on **PS-2(SC)** for access structure $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$: (a) Secret image, (b-i) cover shares of servers Ser_1, Ser_2, Ser_3 and Ser_4, (j) Reconstructed output without any renewal, (k-r) cover shares of servers after second renewal, (s) Reconstructed output at time $t = 2$, (t) Cover image

Ser_2 will do the following.

1) In the case of Γ_1, after adding cover Cov to random shares it becomes, $A_1 = \begin{bmatrix}000 & 111\end{bmatrix}$ and $B_1 = \begin{bmatrix}000 & 111\end{bmatrix}$. Then send $Csh^1_{(2,1,1)}[0] = A_1$ to Ser_1, store $Csh^1_{(2,2,1)}[0] = B_1$ and send $Csh^1_{(2,3,1)}[0] = B_1$ to Ser_3 respectively.
2) In the case of Γ_2, after adding cover Cov to random shares it becomes, $A_2 = \begin{bmatrix}010 & 101\end{bmatrix}$, $A_3 = \begin{bmatrix}010 & 111\end{bmatrix}$ and $B_3 = \begin{bmatrix}000 & 111\end{bmatrix}$.Then store $Csh^1_{(2,2,2)}[0] = A_2$, send $Csh^1_{(2,3,2)}[0] = A_3$ to Ser_3 and $Csh^1_{(2,4,1)}[0] = B_3$ to Ser_4 respectively.

Similarly Ser_3 and Ser_4 will do the same process as mentioned above. Now each server Ser_u for $1 \le u \le n$ can do the share update process as shown below.

Ser_1 will do the following.

1) If qualified set $B \in \Gamma_{QM}$ contains odd number of servers calculate $\mathtt{othS}^1_{(1,1)}[0] = sh^1_{(2,1,1)}[0] \oplus sh^1_{(3,1,1)}[0] \oplus sh^1_{(4,1,1)}[0]$, otherwise calculate $\mathtt{othS}^1_{(1,1)}[0] = sh^1_{(2,1,1)}[0] \oplus sh^1_{(3,1,1)}[0] \oplus sh^1_{(4,1,1)}[0] \oplus (M \odot Ecov)$
2) Outputs $Ush^{(1)}_{(1,1)} \longleftarrow Ush^0_{(1,1)} \oplus sh^1_{(1,1,1)}[0] \oplus \mathtt{othS}^1_{(1,1)}[0]$.

Ser_2 will do the following.

1) If qualified set $B \in \Gamma_{QM}$ contains odd number of servers calculate $\mathtt{othS}^1_{(2,1)}[0] = sh^1_{(1,2,1)}[0] \oplus sh^1_{(3,2,1)}[0] \oplus sh^1_{(4,2,1)}[0]$ and $\mathtt{othS}^1_{(2,2)}[0] = sh^1_{(1,2,2)}[0] \oplus sh^1_{(3,2,2)}[0] \oplus sh^1_{(4,2,2)}[0]$, otherwise calculate $\mathtt{othS}^1_{(2,1)}[0] = sh^1_{(1,2,1)}[0] \oplus sh^1_{(3,2,1)}[0] \oplus sh^1_{(4,2,1)}[0] \oplus (M \odot Ecov)$ and $\mathtt{othS}^1_{(2,2)}[0] = sh^1_{(1,2,2)}[0] \oplus sh^1_{(3,2,2)}[0] \oplus sh^1_{(4,2,2)}[0] \oplus (M \odot Ecov)$.
2) Outputs $Ush^{(1)}_{(2,1)} \longleftarrow Ush^0_{(2,1)} \oplus sh^1_{(2,2,1)}[0] \oplus \mathtt{othS}^1_{(2,1)}[0]$ and $Ush^{(1)}_{(2,2)} \longleftarrow Ush^0_{(2,2)} \oplus sh^1_{(2,2,2)}[0] \oplus \mathtt{othS}^1_{(2,2)}[0]$.

Similarly Ser_3 and Ser_4 will compute and output shares. If qualified set $B \in \Gamma_{QM}$ contains even number of servers, reconstructed image *RI* is generated jointly by

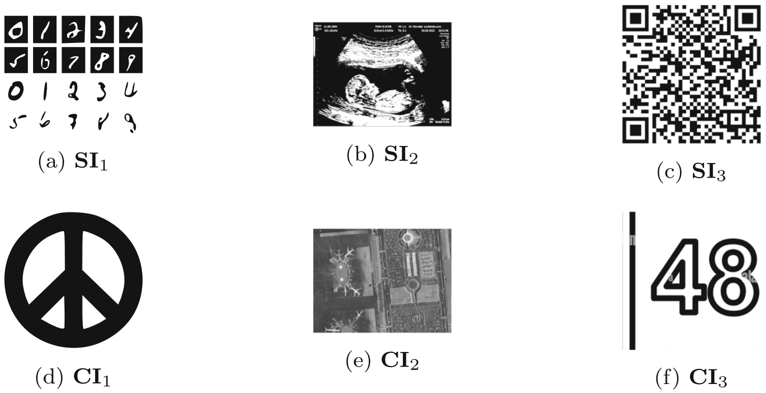
1) Servers Ser_1 and Ser_2 by doing: $RI = Ush^1_{(1,1)} \oplus Ush^1_{(2,1)}$.
2) Servers Ser_1 and Ser_3 by doing: $RI = Ush^1_{(1,1)} \oplus Ush^1_{(3,1)}$.

If qualified set $B \in \Gamma_{QM}$ contains odd number of servers, reconstructed image *RI* is generated jointly by Ser_2, Ser_3 and Ser_4 by doing: $RI = Ush^1_{(2,2)} \oplus Ush^1_{(3,2)} \oplus Ush^1_{(4,1)} \oplus (M \odot Ecov)$.

The average pixel expansion of the scheme is APE $= \frac{3+(3+3)+(3+3)+3}{4} + 3 + 3 = 10.5$.

4.1.6 Experimental results on different Image quality metrics

The binary secret images and the binary cover images of size 1000×753 are shown in Fig. 8. The size of the random shares and meaningful shares (before and after renewals) are 1000×753 and 3000×753 respectively. All the images contain only two grey levels 0 and 255. The **(Entropy)** values corresponding to random shares, meaningful shares and meaningful shares after renewals is given in Table 6. It is clear from Table 6, the entropy values of random shares(sh) > meaningful shares (Csh and $UCsh$) > cover shares ($Ecov$). The peak signal-to-noise ratio **(PSNR)** and structural similarity index measure **(SSIM)** values measured after each renewal while embedding the shares of the secret images in to



(a) $\mathbf{SI}_1$ (b) $\mathbf{SI}_2$ (c) $\mathbf{SI}_3$

(d) $\mathbf{CI}_1$ (e) $\mathbf{CI}_2$ (f) $\mathbf{CI}_3$

Fig. 8 Secret images(a-c) is embedded in the cover images(d-f) respectively

the corresponding cover images is given in Table 7. The correlation coefficients **(Diagonal, Vertical, Horizontal)** values corresponding to random shares, meaningful shares and meanigful shares after renewals is given in Table 8. It is clear from Table 8, the correlation coefficients values of random shares(sh) $<$ meaningful shares (Csh and $UCsh$) $<$ cover shares ($Ecov$).

4.2 Proactive XVCS with different covers for general access structure (PS-2(DC))

In the case of **PS-2(DC)**, meaningful share generation procedure, and reconstruction procedure is the same as given in Section 4.1. In Section 4.1, same cover image Cov is used to generate meaningful shares corresponding to the servers. Then in the case of the share renewal process and reconstruction process, servers are utilizing the expanded cover image $Ecov$ given by the dealer. But it is possible to use multiple cover images Cov_m to generate meaningful shares corresponding to the servers. In this case also, during the share renewal process and reconstruction process, servers are utilizing the expanded cover images $Ecov_m$ given by the dealer. Due to the usage of multiple cover images, the APE will be more compared to the scheme provided in Section 4.1. In this section, we extended the scheme **PS-2** in Section 3.2 by adding multiple covers to the random shares to generate meaningful shares. A schematic description of the protocol is given in Fig. 9.

APE of this scheme **PS-2(DC)**= 2×3×APE(**PS-2**) + 3. This scenario is explained in the Example 6. Tables 4 and 5 shows the APE of **PS-2(DC)** for some access structures. For $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$ shares generated using **PS-2(DC)**, after first renewal for some of the qualified subsets (eg: $\{Ser_1, Ser_2, Ser_3\}$ and $\{Ser_1, Ser_3, Ser_4\}$) is shown in Figs. 1 and 2 (see Section 1). The reconstructed output is same as secret MRI image when the qualified subsets of servers (eg: $\{Ser_1, Ser_2, Ser_3\}$) combine their corresponding shares (eg: $UCsh^1_{(1,1)} \oplus UCsh^1_{(2,1)} \oplus UCsh^1_{(3,1)}$). As per step construction by Liu et al. developed using XOR operation it is possible to divide Γ into two parts $\Gamma_1 = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}\}$ and $\Gamma_2 = \{\{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$. So we can use one cover image for Γ_1 and other cover image for Γ_2.

Table 6 The **Entropy** of shares (generated from secret image - **SI**) and cover shares (generated by embedding shares in cover image - **CI**) for $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$

Shares	Image Size	(SI_1, CI_1)	(SI_2, CI_2)	(SI_3, CI_3)
$sh_{(1,1)}$	1000× 753	1	1	1
$sh_{(1,2)}$	1000× 753	1	1	1
$sh_{(2,1)}$	1000× 753	1	1	1
$sh_{(2,2)}$	1000× 753	1	1	1
$sh_{(3,1)}$	1000× 753	1	1	1
$sh_{(3,2)}$	1000× 753	1	1	1
$sh_{(4,1)}$	1000× 753	1	1	1
$sh_{(4,2)}$	1000× 753	1	1	1
$Csh_{(1,1)}$	3000× 753	0.9998	0.9636	0.9041
$Csh_{(1,2)}$	3000× 753	0.9997	0.9632	0.9033
$Csh_{(2,1)}$	3000× 753	0.9997	0.9634	0.9035
$Csh_{(2,2)}$	3000× 753	0.9997	0.9632	0.9033
$Csh_{(3,1)}$	3000× 753	0.9997	0.9634	0.9034
$Csh_{(3,2)}$	3000× 753	0.9997	0.9635	0.9034
$Csh_{(4,1)}$	3000× 753	0.9997	0.9634	0.9034
$Csh_{(4,2)}$	3000× 753	0.9997	0.9636	0.9031
$UCsh^1_{(1,1)}$	3000× 753	0.9997	0.9633	0.9033
$UCsh^1_{(1,2)}$	3000× 753	0.9998	0.9636	0.9031
$UCsh^1_{(2,1)}$	3000× 753	0.9997	0.9634	0.9025
$UCsh^1_{(2,2)}$	3000× 753	0.9998	0.9636	0.9031
$UCsh^1_{(3,1)}$	3000× 753	0.9997	0.9634	0.9034
$UCsh^1_{(3,2)}$	3000× 753	0.9997	0.9635	0.9034
$UCsh^1_{(4,1)}$	3000× 753	0.9997	0.9634	0.9034
$UCsh^1_{(4,2)}$	3000× 753	0.9998	0.9633	0.9032
$UCsh^2_{(1,1)}$	3000× 753	0.9996	0.9634	0.9026
$UCsh^2_{(1,2)}$	3000× 753	0.9997	0.9635	0.9032
$UCsh^2_{(2,1)}$	3000× 753	0.9996	0.9633	0.9036
$UCsh^2_{(2,2)}$	3000× 753	0.9997	0.9635	0.9032
$UCsh^2_{(3,1)}$	3000× 753	0.9997	0.9633	0.9040
$UCsh^2_{(3,2)}$	3000× 753	0.9997	0.9634	0.9038
$UCsh^2_{(4,1)}$	3000× 753	0.9997	0.9633	0.9040
$UCsh^2_{(4,2)}$	3000× 753	0.9998	0.9633	0.9032
$Ecov$	3000× 753	0.9994	0.9168	0.7755

Example 6 Let $Ser = \{Ser_1, Ser_2, Ser_3, Ser_4\}$, SI $= \begin{bmatrix}1 & 0\end{bmatrix}$ and Cov_m denotes the set of servers , secret image and m cover images respectively. Let the minimal qualified set is $\Gamma_{QM} = \{\{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}, \{Ser_2, Ser_3, Ser_4\}\}$. Then as per step construction by Liu et al. developed on XOR operation it is possible to divide Γ_{QM} into two parts $\Gamma_1 = \{Ser_1, Ser_2\}, \{Ser_1, Ser_3\}\}$ and $\Gamma_2 = \{\{Ser_2, Ser_3, Ser_4\}\}$. Then dealer will do the following.

1) In the case of Γ_1, $Ser_2 \sim Ser_3$, so Implement (2, 2)-scheme on *SI* to spawn two shares $A_1 = \begin{bmatrix}0 & 1\end{bmatrix}$ and $B_1 = \begin{bmatrix}1 & 1\end{bmatrix}$. Then distribute A_1 to Ser_1 and B_1 to both Ser_2 and Ser_3 respectively.

Table 7 **PSNR** (in dBs), **SSIM** & relative contrast of cover shares concerning (secret images, cover images) for $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$

Cover shares	(SI_1, CI_1)	(SI_2, CI_2)	(SI_3, CI_3)
$Csh_{(1,1)}$	(55.89, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.89, 0.9938, 2/3)
$Csh_{(1,2)}$	(55.88, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.90, 0.9938, 2/3)
$Csh_{(2,1)}$	(55.88, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.90, 0.9938, 2/3)
$Csh_{(2,2)}$	(55.88, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.90, 0.9938, 2/3)
$Csh_{(3,1)}$	(55.94, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.89, 0.9938, 2/3)
$Csh_{(3,2)}$	(55.91, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$Csh_{(4,1)}$	(55.94, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.89, 0.9938, 2/3)
$Csh_{(4,2)}$	(55.89, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.90, 0.9938, 2/3)
$UCsh^1_{(1,1)}$	(55.93, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^1_{(1,2)}$	(55.90, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^1_{(2,1)}$	(55.90, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^1_{(2,2)}$	(55.90, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^1_{(3,1)}$	(55.92, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.92, 0.9939, 2/3)
$UCsh^1_{(3,2)}$	(55.88, 0.9936, 2/3)	(55.90, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^1_{(4,1)}$	(55.92, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.92, 0.9939, 2/3)
$UCsh^1_{(4,2)}$	(55.91, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.93, 0.9939, 2/3)
$UCsh^2_{(1,1)}$	(55.95, 0.9938, 2/3)	(55.91, 0.9939, 2/3)	(55.92, 0.9938, 2/3)
$UCsh^2_{(1,2)}$	(55.87, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^2_{(2,1)}$	(55.91, 0.9937, 2/3)	(55.91, 0.9939, 2/3)	(55.89, 0.9938, 2/3)
$UCsh^2_{(2,2)}$	(55.87, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^2_{(3,1)}$	(55.95, 0.9937, 2/3)	(55.92, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^2_{(3,2)}$	(55.91, 0.9937, 2/3)	(55.92, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^2_{(4,1)}$	(55.95, 0.9937, 2/3)	(55.92, 0.9939, 2/3)	(55.91, 0.9938, 2/3)
$UCsh^2_{(4,2)}$	(55.89, 0.9936, 2/3)	(55.91, 0.9939, 2/3)	(55.91, 0.9938, 2/3)

2) In the case of Γ_2, again Implement (2, 2)-scheme on *SI* to spawn two new shares $A_2 = \begin{bmatrix}0 & 0\end{bmatrix}$ and $B_2 = \begin{bmatrix}1 & 0\end{bmatrix}$. Implement (2, 2)-scheme on B_2 to spawn shares $A_3 = \begin{bmatrix}1 & 1\end{bmatrix}$ and $B_3 = \begin{bmatrix}0 & 1\end{bmatrix}$.Then distribute A_2 to Ser_2, A_3 to Ser_3 and B_3 to Ser_4 respectively. In the following, we elaborate on the share generation and embedding details.

1. Ser_1 holds share $sh_{(1,1)} = A_1$ and this will be embedded in the cover image Cov_1 to generate $Csh_{(1,1)}$.
2. Ser_2 holds share $sh_{(2,1)} = B_1$ and this will be embedded in the cover image Cov_1 to generate $Csh_{(2,1)}$.
3. Ser_2 holds one more share $sh_{(2,2)} = A_2$ and this will be embedded in the cover image Cov_2 to generate $Csh_{(2,2)}$.
4. Ser_3 holds shares $sh_{(3,1)} = B_1$ and this will be embedded in the cover image Cov_1 to generate $Csh_{(3,1)}$.
5. Ser_3 holds one more share $sh_{(3,2)} = A_3$ and this will be embedded in the cover image Cov_2 to generate $Csh_{(3,2)}$.
6. Ser_4 holds share $sh_{(4,1)} = B_3$ and this will be embedded in the cover image Cov_2 to generate $Csh_{(4,1)}$.

Table 8 The correlation table **(Diagonal, Vertical, Horizontal)** of shares (generated from secret image - **SI**) and cover shares (generated by embedding shares in cover image - **CI**) for $\Gamma = \{\{Ser_1, Ser_2, Ser_3\}, \{Ser_1, Ser_2, Ser_4\}, \{Ser_1, Ser_3, Ser_4\}, \{Ser_2, Ser_3, Ser_4\}\}$

Shares	(SI_1, CI_1)	(SI_2, CI_2)	(SI_3, CI_3)
$sh_{(1,1)}$	(0.0042, -0.0095, 0.0033)	(0.0011, 0.0009, 0.0018)	(-0.0011, -0.0001, 0.0010)
$sh_{(1,2)}$	(0.0055, -0.0062, -0.0040)	(-0.0003, 0.0014, 0.0018)	(0.0027, 0.0003, -0.0002)
$sh_{(2,1)}$	(-0.0009, -0.0047, 0.0027)	(0.0005, 0.0021, -0.0009)	(0.0013, 0.0007, -0.0001)
$sh_{(2,2)}$	(0.0055,-0.0062, -0.0040)	(-0.0003, 0.0014, 0.0018)	(0.0027, 0.0003, -0.0002)
$sh_{(3,1)}$	(0.0037,-0.0057, 0.0015)	(-0.0012, -0.0023, -0.0014)	(-0.0003, 0.0029, 0.0024)
$sh_{(3,2)}$	(-0.0043,-0.0080, 0.0066)	(-0.0028, 0.0003, -0.0004)	(0.0019, 0.0003, -0.0033)
$sh_{(4,1)}$	(0.0037,-0.0057, 0.0015)	(-0.0012, -0.0023, -0.0014)	(-0.0003, 0.0029, 0.0024)
$sh_{(4,2)}$	(0.0046,-0.0117, 0.0061)	(-0.0007, 0.0007, 0.0035)	(-0.0002, 0.0017, 0.0010)
$Csh_{(1,1)}$	(0.4175, 0.4247, 0.3595)	(0.3306, 0.3359, 0.3115)	(0.3424, 0.3479, 0.2675)
$Csh_{(1,2)}$	(0.4181, 0.4209, 0.3581)	(0.3304, 0.3360, 0.3112)	(0.3424, 0.3481, 0.2669)
$Csh_{(2,1)}$	(0.4221, 0.4196, 0.3588)	(0.3313, 0.3366, 0.3111)	(0.3437, 0.3480, 0.2671)
$Csh_{(2,2)}$	(0.4207, 0.4225,0.3568)	(0.3304, 0.3356, 0.3115)	(0.3412, 0.3479, 0.2679)
$Csh_{(3,1)}$	(0.4235, 0.4304, 0.3661)	(0.3312, 0.3352, 0.3110)	(0.3425, 0.3484, 0.2666)
$Csh_{(3,2)}$	(0.4201, 0.4303, 0.3634)	(0.3315, 0.3359, 0.3111)	(0.3462, 0.3486, 0.2690)
$Csh_{(4,1)}$	(0.4241, 0.4299, 0.3646)	(0.3310, 0.3360, 0.3111)	(0.3421, 0.3476, 0.2665)
$Csh_{(4,2)}$	(0.4206, 0.4263,0.3579)	(0.3308, 0.3357, 0.3111)	(0.3432, 0.3477, 0.2664)
$UCsh^1_{(1,1)}$	(0.4216, 0.4274, 0.3655)	(0.3302, 0.3352, 0.3108)	(0.3447, 0.3497, 0.2699)
$UCsh^1_{(1,2)}$	(0.4232, 0.4241, 0.3629)	(0.3310, 0.3364, 0.3117)	(0.3439, 0.3478, 0.2695)
$UCsh^1_{(2,1)}$	(0.4231, 0.4265, 0.3597)	(0.3316, 0.3368, 0.3118)	(0.3439, 0.3484, 0.2684)
$UCsh^1_{(2,2)}$	(0.4281, 0.4250, 0.3617)	(0.3314, 0.3361, 0.3118)	(0.3450, 0.3477, 0.2689)
$UCsh^1_{(3,1)}$	(0.4249, 0.4273, 0.3636)	(0.3322, 0.3364, 0.3112)	(0.3454, 0.3500, 0.2713)
$UCsh^1_{(3,2)}$	(0.4190, 0.4255, 0.3579)	(0.3302, 0.3352, 0.3107)	(0.3431, 0.3491, 0.2691)
$UCsh^1_{(4,1)}$	(0.4242, 0.4300, 0.3656)	(0.3314, 0.3362, 0.3109)	(0.3454, 0.3473, 0.2712)
$UCsh^1_{(4,2)}$	(0.4257, 0.4257, 0.3629)	(0.3317, 0.3363, 0.3117)	(0.3469, 0.3507, 0.2714)
$UCsh^2_{(1,1)}$	(0.4279, 0.4317, 0.3679)	(0.3318, 0.3361, 0.3126)	(0.3447, 0.3492, 0.2707)
$UCsh^2_{(1,2)}$	(0.4218, 0.4209, 0.3588)	(0.3319, 0.3362, 0.3123)	(0.3444, 0.3482, 0.2696)
$UCsh^2_{(2,1)}$	(0.4250, 0.4264, 0.3624)	(0.3319, 0.3372, 0.3123)	(0.3421, 0.3463, 0.2667)
$UCsh^2_{(2,2)}$	(0.4183, 0.4232, 0.3571)	(0.3316, 0.3360, 0.3128)	(0.3445, 0.3490, 0.2699)
$UCsh^2_{(3,1)}$	(0.4294, 0.4303, 0.3685)	(0.3320, 0.3375, 0.3126)	(0.3441, 0.3500, 0.2692)
$UCsh^2_{(3,2)}$	(0.4232, 0.4292, 0.3639)	(0.3317, 0.3374, 0.3122)	(0.3449, 0.3488, 0.2704)
$UCsh^2_{(4,1)}$	(0.4222, 0.4326, 0.3683)	(0.3319, 0.3367, 0.3126)	(0.3443, 0.3486, 0.2694)
$UCsh^2_{(4,2)}$	(0.4175, 0.4254, 0.3600)	(0.3325, 0.3361, 0.3118)	(0.3436, 0.3498, 0.2703)
$Ecov$	(0.9511, 0.9598, 0.9818)	(0.7976, 0.8095, 0.9381)	(0.9531, 0.9657, 0.9815)

Here two cover images ($m = 2$) used are Cov_1 and Cov_2. So dealer need to share the following.

1. $Ecov_1$ and $Csh_{(1,1)}$ is shared with Ser_1.
2. $Ecov_1$, $Ecov_2$, $Csh_{(2,1)}$ and $Csh_{(2,2)}$ is shared with Ser_2.
3. $Ecov_1$, $Ecov_2$, $Csh_{(3,1)}$ and $Csh_{(3,2)}$ is shared with Ser_3.
4. $Ecov_2$ and $Csh_{(4,1)}$ is shared with Ser_4.

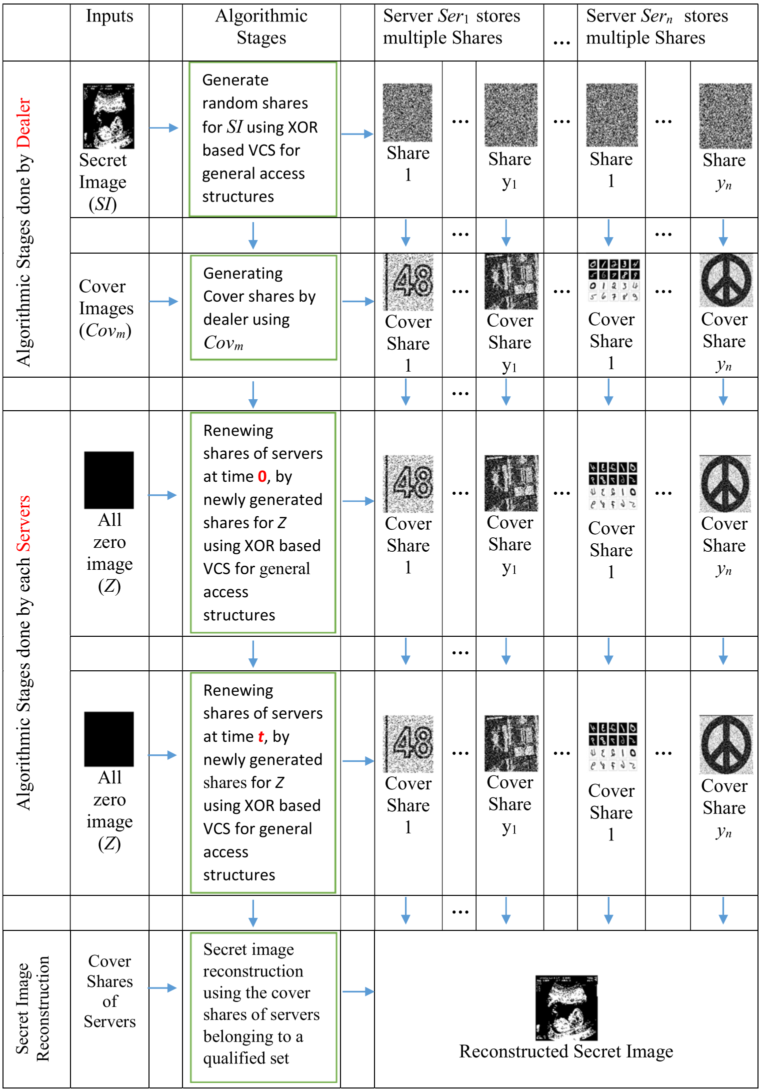

Inputs
Algorithmic Stages
Server Ser_1 stores multiple Shares
...
Server Ser_n stores multiple Shares
Algorithmic Stages done by Dealer
Secret Image (SI)
Generate random shares for SI using XOR based VCS for general access structures
Share 1
Share y_1
Share 1
Share y_n
Cover Images (Cov_m)
Generating Cover shares by dealer using Cov_m
Cover Share 1
Cover Share y_1
Cover Share 1
Cover Share y_n
Algorithmic Stages done by each Servers
All zero image (Z)
Renewing shares of servers at time 0, by newly generated shares for Z using XOR based VCS for general access structures
Renewing shares of servers at time t, by newly generated shares for Z using XOR based VCS for general access structures
Secret Image Reconstruction
Cover Shares of Servers
Secret image reconstruction using the cover shares of servers belonging to a qualified set
Reconstructed Secret Image

Fig. 9 Algorithmic stages of our construction(**PS-2(DC)**): DC stands for different covers

5. Permutation share M is shared with all servers Ser_1, Ser_2, Ser_3 and Ser_4.

Since all the shares have pixel expansion 3, then APE $= 2 \times 3 \times (1 + 2 + 2 + 1)/4 + 3$.

5 Conclusions

Updating the secret states of servers at a regular time interval is essential for long-term storage of confidential data. If the data is in the format of an image then introducing refreshing techniques into the visual cryptographic scheme poses some challenges.Towards this, we propose proactive visual cryptographic schemes for the XOR-based schemes. Our proposal is information theoretically secure with provable security and thus provides security even against quantum computers. The updating procedure can be performed an unlimited number of times without degrading the quality of the secret image. The robustness of the proposed constructions are also validated using the Image quality metrics like PSNR, SSIM, Correlation coefficients and Entropy.

Acknowledgements The research of Sabyasachi Dutta is supported by MITACS Accelerate fellowship, Canada vide Ref. No. IT25625, FR66861. The research of Avishek Adhikari is partially supported by DST-SERB Project MATRICS vide Sanction Order: MTR/2019/001573 and the DST-FIST Project SR/FST/MS-I/2019/41.

Data Availability Due to the sensitive nature of the data, information created during and/or analysed during the current study is available from the corresponding author [Praveen. K, *k_praveen@cb.amrita.edu*] on reasonable request to researchers.

Declarations

Conflict of Interests The authors declare that they do not have any conflict of interest/competing interests.

References

1. Adhikari A (2014) Linear algebraic techniques to construct monochrome visual cryptographic schemes for general access structure and its applications to color images. Des Codes Cryptograph 73(3):865–895
2. Adhikari A, Dutta TK, Roy BK (2004) A new black and white visual cryptographic scheme for general access structures. In: Canteaut A, Viswanathan K (eds) Progress in cryptology - INDOCRYPT 2004, 5th international conference on cryptology in India, Chennai, India, December 20-22, 2004, proceedings, volume 3348 of Lecture notes in computer science, Springer, pp 399–413
3. Aggarwal AK (2022) Biological tomato leaf disease classification using deep learning framework. Int J Biol Biomed Eng 16:241–244
4. Aggarwal AK (2022) Learning texture features from glcm for classification of brain tumor mri images using random forest classifier. J Wseas Trans Signal Process 60–63
5. Ateniese G, Blundo C, DeSantis A, Stinson DR (1996) Visual cryptography for general access structures. Inf Comput 129(2):86–106
6. Ateniese G, Blundo C, DeSantis A, Stinson DR (2001) Extended capabilities for visual cryptography. Theor Comput Sci 250(1-2):143–161
7. Chopra J, Kumar A, Aggarwal AK, Marwaha A (2018) An efficient watermarking for protecting signature biometric template. In: 2018 5th International conference on signal processing and integrated networks (SPIN), IEEE, pp 413–418
8. Chuang T, Chen C, Chien B (2016) Image sharing and recovering based on chinese remainder theorem. In: 2016 International symposium on computer, Consumer and Control (IS3C), pp 817–820
9. Dutta S, Adhikari A (2014) Xor based non-monotone $t-(k,n)^*$ -visual cryp tographic schemes using linear algebra. In: International conference on information and communications security, Springer, pp 230–242
10. Dutta S, Adhikari A, Ruj S (2019) Maximal contrast color visual secret sharing schemes. Codes Cryptogr 87(7):1699–1711
11. Dutta S, Rohit R, Adhikari A (2016) Constructions and analysis of some efficient $t-(k,n)^*$ - visual cryptographic schemes using linear algebraic techniques. Des Codes Crypt 80(1):165–196
12. Dutta S, Sardar M, Adhikari A, Ruj S, Sakurai K (2020) Color visual cryptography schemes using linear algebraic techniques over rings. In: International conference on information systems security, Springer, pp 198–217
13. Espejel-Trujillo A, Iwamoto M, Nakano-Miyatake M (2018) A proactive secret image sharing scheme with resistance to machine learning based steganalysis. Multimed Tools Appl 77(12):15161–15179
14. Guo C, Yuan Q, Lu K, Li M, Fu Z (2017) (t, n) threshold secret image sharing scheme with adversary structure. Multimed Tools Appl 76(20):21193–21210

15. Guo C, Zhang H, Zhangjie F, Feng B, Li M (2018) A novel proactive secret image sharing scheme based on liss. Multimed Tools Appl 77(15):19569–19590
16. Herzberg A, Jarecki S, Krawczyk H, Yung M (1995) Proactive secret sharing or: how to cope with perpetual leakage. In: Annual international cryptology conference, Springer, pp 339–352
17. Kanakkath P, Madathil S, Krishnan R (2019) Deterministic extended visual cryptographic schemes for general access structures with or-and and xor-and operations. Multimed Tools Appl 78(2):1315–1344
18. Kaur A, Chauhan AP, Aggarwal AK (2022) Dynamic deep genomics sequence encoder for managed file transfer. IETE J Res 1–13
19. Khan A, Li JP, Haq A, Memon I, Patel S et al (2021) Emotional-physic analysis using multi-feature hybrid classification. J Intell Fuzzy Syst 40(1):1681–1694
20. Li L, Lu Y, Yan X, Liu L, Tan L (2019) Lossless (k,n) -threshold image secret sharing based on the chinese remainder theorem without auxiliary encryption. IEEE Access 7:75113–75121
21. Liu F, Wu C, Lin X (2009) Step construction of visual cryptography schemes. IEEE Trans Inf Forensic Secur 5(1):27–38
22. Longdan T, Lu Y, Yan X, Liu L, Li L (2019) Weighted secret image sharing for a (k,n) threshold based on the chinese remainder theorem. IEEE Access 05:1–1
23. Ma C, Ding X (2009) Proactive verifiable linear integer secret sharing scheme. In: International conference on information and communications security, Springer, pp 439–448
24. Naor M, Shamir A (1995) Visual cryptography. In: DeSantis A (ed) Advances in Cryptology — EUROCRYPT'94. Springer, Berlin, pp 1–12
25. Ostrovsky R, Yung M (1991) How to withstand mobile virus attacks. In: Proceedings of the tenth annual ACM symposium on Principles of distributed computing, pp 51–59
26. Praveen K, Sethumadhavan M (2018) Blind authentication based cheating immune xor step construction for visual cryptography. Int J Pure Appl Math 118(18):2847–2854
27. Praveen K, Indu G, Santhya R, Sethumadhavan M (2017) An android application for secret image sharing with cloud storage, pp International symposium on security in computing and communication, Springer, pp 399–410
28. Praveen K, Sethumadhavan M (2016) Ideal contrast visual cryptography for general access structures with and operation
29. Praveen K, Sethumadhavan M (2017) On the extension of xor step construction for optimal contrast grey level visual cryptography. In: 2017 International conference on advances in computing, communications and informatics (ICACCI), IEEE, pp 219–222
30. Sardar MK, Adhikari A (2020) Essential secret image sharing scheme with small and equal sized shadows. Signal Process Image Commun 87:115923
31. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
32. Srinivasu PN, NORWAWİ N, Amiripalli SS, Deepalakshmi P Secured compression for 2d medical images through the manifold and fuzzy trapezoidal correlation function. Gazi University Journal of Science, pp 1–1
33. Thien C-C, Lin J-C (2002) Secret image sharing. Comput Graph 26:765–770 10
34. Tunio MH, Jianping L, Butt MHF, Memon I (2021) Identification and classification of rice plant disease using hybrid transfer learning. In: 2021 18th International computer conference on wavelet active media technology and information processing (ICCWAMTIP), IEEE, pp 525–529
35. Tuyls Pim, Hollmann HenkDL, VanLint JackH, LMGM Tolhuizen. (2005) Xor-based visual cryptography schemes. Des Codes Crypt 37(1):169–186
36. Ubhi JS, Aggarwal AK et al (2022) Neural style transfer for image within images and conditional gans for destylization. J Vis Commun Image Represent 85:103483
37. Verheul ER, Van Tilborg HCA (1997) Constructions and properties of k out of n visual secret sharing schemes. Des Codes Crypt 11(2):179–196
38. Wang G, Kang W, Wu Q, Wang Z, Gao J (2018) Generative adversarial network (gan) based data augmentation for palmprint recognition. In: 2018 Digital image computing: techniques and applications (DICTA), IEEE, pp 1–7
39. Yan X, Lu Y, Liu L, Wan S, Ding W, Liu H (2017) Chinese remainder theorem-based secret image sharing for (k,n) threshold. In: Sun X, Chao H-C, You X, Bertino E (eds) Cloud Computing and Security. Springer, Cham, pp 433–440
40. Yang C-N, Laih C-S (2000) New colored visual secret sharing schemes. Des Codes Crypt 20(3):325–336
41. Yang C-N, Wang D-S (2013) Property analysis of xor-based visual cryptography. IEEE Trans Circ Syst Video Technol 24(2):189–197

42. Yuqiao C, Zhengxin F, Bin Y (2023) A xor-based visual cryptography scheme for (2, n) access structure with ideal structure division. J Syst Simul 32(1):20
43. Zhou Z, Yang C, Sun X (2018) Secret image sharing based on encrypted pixels. IEEE Access 6:15021–15025

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.