IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 June 2024, accepted 17 July 2024, date of publication 22 July 2024, date of current version 30 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3431717

RESEARCH ARTICLE

A Self-Authenticating Multi-Tone Secret Sharing Scheme Using Meaningful Shares for Satellite Images

SUCHITA SHARMA, SHIVENDRA SHIVANI, AND NITIN SAXENA
CSED, Thapar Institute of Engineering & Technology, Patiala, Punjab 147004, India
Corresponding author: Suchita Sharma (ssharma60_phd20@thapar.edu)

ABSTRACT In today's era of far-reaching usage of satellite imagery across diverse fields such as remote sensing, environmental monitoring, and national security, the authenticity and integrity of these images have become paramount. However, ensuring the trustworthiness of satellite images remains a challenge, particularly in scenarios where unauthorized access or tampering can lead to severe consequences. This paper proposes a novel solution to address this challenge by means of a self-authenticating multi-tone secret sharing visual cryptography (VC) scheme with meaningful shares. The proposed approach incorporates novel processes, i.e., SPICE (secure pixel integration for covert embedding) and EX-CEP (extraction of covertly embedded pixels) for generation and extraction of secret shares at the sender and receiver side, respectively. Prior to share generation, the satellite image is encrypted and processed to generate a secret hash value which is used for establishing the authenticity and integrity of the received satellite image. The presented work delineates an innovative approach which integrates a comprehensive set of features, i.e., Image Encryption, Self-Authentication, Integrity Verification, Meaningful Shares Construction (from full 8-bit grayscale Meaningless Shares), 100% Original Secret Reconstruction, No Contrast Loss, No Share Expansion, under one umbrella. This approach is distinguished by its holistic one-stop solution strategy, which seamlessly amalgamates multiple novel techniques into a singular, cohesive framework. Exhaustive set of experiments have been performed to demonstrate the efficacy of the proposed approach and to substantiate it's superiority over the existing state-of-the-art VC approaches. Through comprehensive experimentation, the presented work underscores the effectiveness of the proposed scheme in safeguarding sensitive satellite imagery.

INDEX TERMS Visual cryptography, multi-tone images, meaningful shares, remote sensing, multiple shares.

I. INTRODUCTION

In today's rapidly evolving world, the demand for securing satellite images has become more pressing than ever. These images offer a unique vantage point, allowing us to monitor and understand our planet in unprecedented ways ([1], [2], [3], [4], [5], [6]). Whether be it tracking environmental changes [1], assessing natural disasters [2], assisting in urban planning [4] or their role in safeguarding our national security [6], satellite images are an invaluable resource. In the context of national security, securing satellite images is of utmost necessity because they are used to gather intelligence on foreign adversaries, their hidden criminal training camps or armed forces base stations, to track the movement of military forces, to identify and track threats to critical infrastructure, such as power plants, water treatment facilities, and transportation networks, etc. [6]. Securing satellite images can help to protect such sensitive information from falling into the wrong hands who can misuse it to attack our nation's critical infrastructure, to target people's privacy with unauthorized intrusion, etc. In context of environmental conservation, securing satellite images is essential for

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

tracking the effects of climate change. These images help scientists and researchers study the shrinkage of ice caps, deforestation, and other critical environmental indicators [1], [3], [4]. Furthermore, satellite images play a pivotal role in disaster management, allowing authorities to assess the impact of natural calamities like hurricanes, earthquakes, and wildfires in real-time [2]. This timely information can save lives and aid in disaster relief efforts. Thus, it is very imperative to secure the information content of satellite images and employ stringent security measures to prevent their unauthorized access and to detect even a minutest of tampering during their transmission. In this regard, an attempt has been made to propose a self-authenticating multi-tone secret sharing VC approach for securing satellite images using meaningful shares.

Visual cryptography is a secret sharing scheme that turns ordinary images into cryptic puzzles, ensuring secure communication and data protection [7]. VC works by dividing the secret image into two or more shares, where each share is a random-looking image (resembling noise) that contains no information about the secret image on their own. However, when the shares are combined, the secret image is revealed. VC can also ensure access control such that every authorized party will have a piece of the information in the form of a share, and only when enough (or all) share images are combined together, the hidden content gets unveiled [7], [8]. No individual party (in case it got rogue or compromised) can deduce the hidden information on their own with single share. These advantages has enabled the wide spread use of VC in numerous applications such as remote sensing, secure banking communication, anti-phishing systems, defense systems, etc. [9], [10]

Traditional VC schemes were mainly computation-less schemes targeted towards securing binary images. These schemes suffered from major drawbacks, i.e., pixel expansion and contrast loss. Also, traditional VC schemes tend to divide secret image into meaningless (noise like) shares which sometimes causes suspicion in the mind of malicious attacker that a meaningless random share might be concealing some valuable information in itself [10]. Thus, with the advent of technology, a new breed of computation-based VC schemes came into existence which were immune to these drawbacks. Computation-based VC schemes have the advantage of recovering 100% contrast in reconstructed secret image obtained by combining the shares with zero pixel expansion [7]. Also, a growing interest in the use of meaningful shares to conceal secret information in place of meaningless random shares has been observed among such computation-based VC schemes. Meaningful shares [11] are more effective because they carry valid visual information, seems to be natural and very difficult to garner suspicion from attacker's point of view.

The motivation for our work has been drawn from the need of an one-stop solution computation-based VC approach which integrates a comprehensive set of features, i.e., Image Encryption, Self-Authentication, Integrity Verification, Meaningful Shares Construction (from full 8-bit grayscale Meaningless Shares), 100% Original Secret Reconstruction, No Contrast Loss, No Share Expansion, under one umbrella. The existing computation-based VC approaches lack in one or more features listed above, i.e., either relying only on meaningless shares or lack of encryption module before the share generation, or lack of integrity verification/tamper detection module at the receiver end.

In view of this, we have curated our proposed approach as computation-based 3-out-of-3 visual cryptography scheme, where three random shares generated from the original satellite image will be embedded into three natural cover images to form three meaningful shares. Our proposed approach is a special case of n-out-of-n visual secret sharing scheme, where a given secret image is encoded into n shares and actual secret image can only be disclosed by combination of all the n shares. One cannot reveal any data about the secret image by inspecting less than n shares, even if infinite computational power is available. The following points highlights the contributions of our proposed approach:

- **Three Level Encryption:** The proposed approach employs three level encryption strategy to generate encrypted satellite image; Block Scrambling, Pixel Scrambling and S-Box substitution.
- **Self-Authentication Mechanism for Tamper Detection:** The proposed approach embodies an hash value generating mechanism. A comparison of hash values computed at the sender and receiver side substantiates the integrity and authenticity of the received satellite image.
- **Secret Share Generation Using Newton Polynomials:** The proposed approach incorporates the use of Newton basis polynomials to generate three grayscale random (meaningless) shares.
- **Novel Construction of Meaningful Shares:** The proposed approach generates three meaningful shares by embedding each of the three random meaningless shares in a natural cover image using proposed SPICE (Secured Pixel Integration For Covert Embedding) technique.

The rest of the paper is structured as follows: Section II describes the related and existing work done on the security of satellite images. Section III describes the working of our proposed multi-tone VC scheme in detail. To exhibit the effectiveness of the proposed approach, the experimental results and comparisons with various state of the art approaches are discussed in section IV. The conclusions are presented in Section V.

II. RELATED WORK

Naor and Shamir [15] first introduced the VC approach to encrypt visual information. They proposed (k, n) threshold scheme is visual secret sharing (VSS) scheme for sharing visual information over the public network securely. The secret image is split in to n random meaningless shares. The secret image can only be reconstructed by joining no

less than any k shares. The scheme is computation less and suitable for binary images but suffers from pixel expansion and contrast loss problem. Ateniese et al. [16] later proposed enhanced VSS scheme uses general access structure for shares generation. The shares are grouped as legitimate users share and forbidden users share. The secret image can only be decrypt by legitimate users share only and not by forbidden users share. Several other variants of VSS scheme were proposed to improve original VSS scheme [17], [18].

Complexity associated with meaningless or noise like shares leads to the development of schemes generating meaning full shares of secret image. A reversible secret image sharing scheme used Sudoku matrix and Lagrange polynomial to generate meaningful shares is presented in [19]. A turtle shell matrix based scheme with reversibility and authentication is proposed to generate meaningful share [20]. Schemes which use single cover image for generating multiple shares are still difficult to maintain and vulnerable to steganalysis.

Yuan et al. [21] proposed secret image sharing scheme based on multi-cover adaptive steganography for natural images. The secret image bits are adaptively embedded into multiple cover images hence pixel expansion cannot be avoided. Another secret haring scheme based on gray code that generates meaning full shares is proposed in [22]. The authors used absolute moment block truncation Coding (AMBTC) compression to reduce communication overhead. Chiu et al. [11] proposed OR and XOR based $(2, n)$ VC schemes to generate meaningful shares. Further, He et al. [23] applied LOCO-I compression to cut down the statistical correlations between neighbouring pixels of a secret image.

Zhou et al. [24] proposed a (k, n) secret image sharing scheme based on encrypted pixels rather than permutation as used in traditional schemes. A verifiable XOR based visual secret sharing scheme for hyperspectral images is proposed in [25]. The scheme selects appropriate band in the preprocessing step to reduce image size by avoiding redundancy. A (k, n) secret document sharing scheme proposed by Yang et al. [26], ensures easy maintenance of shares by generating meaningful shares. Recently a fractal reference matrix based scheme is presented in [27]. The scheme is an authenticable $(2, 3)$ secret sharing scheme which generates meaningful shares.

Another secret image sharing scheme aimed to reduce distortion in recovered image is proposed by Yang et al. [28]. The scheme uses multi prime modular arithmetic instead of Galois field to reduce computational complexity. On the similar grounds, a meaningful visual cryptography approach for remote sensing images has been presented by Zhang et al. in [29]. Their proposed work is targeted for computation limited IoT devices in IoT-cloud infrastructure. Their proposed VC approach suffers from the degradation in the quality of the reconstructed image, which they have attempted to compensate by diffusing the error between the encryption block and the original block to adjacent blocks. Another VC based approach for IoT-cloud infrastructure [30] has been presented for developing a mutual authentication protocol for accessing the cloud services. The mutual authentication is based on two secret images and tickets, whereby visual cryptography is used to encrypt and decrypt the secret images.

Rawat et al. [31] presented a lightweight (n, n) secret image sharing scheme for medical images. Scheme used Boolean XOR and LSB stuffing as lightweight operation to generate meaningful share. Another VC scheme has been presented recently in [32] for securing the transmission of medical images using progressive meaningful shares. The presented approach in place of encrypting the medical image before share generation encrypts the meaningful shares before sending to the receiver side. This approach also suffers from lossy reconstruction of the original image, which in case of medical images is a huge drawback. Wu and Chen [33] presented a lightweight (n, n) non-expansible XOR based meaningful VC scheme capable of 100% reconstruction of the original image. However, the use of simple XOR operator limits the functioning of the presented work to binary meaningless shares only. Lastly, a comprehensive survey of secret image sharing schemes is presented in [34]. Readers can refer to this survey for greater in-depth into the categorization of secret sharing schemes in several categories such as visual secret sharing, secret image sharing with meaningful shares and discuss their pros and cons.

III. PROPOSED SECRET SHARING APPROACH

The general outline of the proposed approach is demonstrated in Figure 1. A sequence of steps, i.e., Encryption of secret image, Hash value generation, Generation of Meaningless shares and SPICE (Meaningful share generation) as shown in the figure will be performed on sender side to hide the grayscale (satellite) image in three meaningful shares. The receiver will execute the same sequence of steps but in reverse order, i.e., EX-CEP (Meaningless share extraction), Encrypted Secret Extraction, Hash Value Comparison, Decryption, to extract the original grayscale (satellite) image from the three meaningful shares received at it's end. The work flow of each step is sub-divided into smaller steps, which have been elaborated in detail in later subsections.

The proposed work entails the sharing of minutest of supplementary information in the form of three security keys (used in the encryption, hash generation and construction of meaningful shares) and the computed hash value by the sender side with the receiver. The receiver on receiving this supplementary information will be able to effectively extract the meaningless shares from the meaningful ones, which would be processed further for the extraction of original (encrypted) secret image. Once secret is extracted, it will be checked for it's integrity using the received hash value. If the received and recalculated hash values do not match, then the receiver will request for the re-transmission of the shares to the sender side. The proposed approach fulfils three security requirements namely; 1) Confidentiality using complex

encryption and visual cryptography, 2) Authentication via three layers of security using three symmetric keys, and 3) Integrity using hash value. Thus, with the incorporation of such features, the proposed approach is very apt for securing satellite images.

The presented work delineates an innovative approach which exhibits a significant enhancement over the existing paradigms in visual cryptography by integrating a comprehensive set of features, i.e., *Image Encryption, Self-Authentication, Integrity Verification, Meaningful Shares Construction (from full 8-bit grayscale Meaningless Shares), 100% Original Secret Reconstruction, No Contrast Loss, No Share Expansion*, under one umbrella. This approach is distinguished by it's holistic one-stop solution strategy, which seamlessly amalgamates multiple novel techniques into a singular, cohesive framework. Key highlights include:

i) **Enhanced Encryption and Self-Authentication:** At the core of our VC approach is a sophisticated image encryption methodology that employs a tri-layered security mechanism. This not only ensures the confidentiality of the secret image but also facilitates it's self-authentication, thereby substantially elevating the robustness of our proposed approach.
ii) **Integrity Verification Through Advanced Hashing:** The paper introduces an innovative and efficient hashing technique tailored for integrity verification of the secret image. This method is capable of generating a 128-bit hash value, providing a robust mechanism for verifying the integrity of the image without compromising security.
iii) **Efficient Shares Generation:** A novel, straightforward methodology based on Newton basis polynomials is proposed for the generation of unexpanded (full 8-bit grayscale) meaningless shares. This represents a significant leap forward in simplifying the shares generation process while ensuring that the size of the shares remains unaltered, thereby addressing the common issue of share expansion.
iv) **Innovative Conversion to Meaningful Shares:** A novel method, dubbed SPICE (Secured Pixel Integration For Covert Embedding), is presented for the conversion of meaningless shares into meaningful ones. This innovative technique not only enhances the aesthetic appeal of the shares but also contributes to the concealment of the secret image, adding an extra layer of security.
v) **Flawless Secret Image Reconstruction:** A standout feature of this approach is it's capability to accurately recover the original secret image at the receiver's end, with a 100% success rate and no loss in contrast. This ensures that the integrity of the secret image is meticulously preserved throughout the process, from encryption to decryption.

All the intermediate steps are explained in details with supporting figures in following subsections.

A. ENCRYPTION

The first step in ensuring the security of satellite images is image encryption. The encryption process is a sequence of three algorithms, namely; 1) Block scrambling, 2) Pixel scrambling and 3) S-Box value substitution, which provide high level of confidentiality. The use of three symmetric keys $K1, K2$ and $K3$ is employed in these three algorithms to ensure the authenticity of sender. These security keys in our work have been generated from well-known and large key space generators. Using these large key space generators enables our approach to be more robust towards malicious decoding through any man-in-the-middle attack. Using different key generators allows the system to leverage the unique strengths of each generator, addressing a wide range of requirements from high-quality statistical sampling to secure, parallel random number generation. This approach ensures flexibility, robustness, and security. Even with the knowledge of these algorithms, an eves-dropper (or any man-in-the-middle attacker) wouldn't be able to decrypt or extract the contents of the original image without having the possession of $K1, K2$ and $K3$. At the receiver side, the same set of algorithms will decrypt the received image, however, the order of execution of these algorithms will be in reverse, i.e., Inverse S-Box value substitution followed by Pixel scrambling and at last, Block scrambling. Figure 2 shows the complete process of image encryption with the help of a small example for better understanding.

1) BLOCK SCRAMBLING

In this step, the input secret image will be scrambled block-wise. First of all, input image is divided into non-overlapping blocks of size 16×16. Let N represent the count of such non-overlapping 16×16 blocks. Here, one assumption is made that the size of the input secret image will be 512×512. Let, ***blockIndices*** represent a row vector of block indices of size $1 \times N$ of the form: ***blockIndices*** $\in \{1, 2, \ldots, N\}$. In order to randomize the block, a secret key $K1$ is used. This key is generated by Mersenne Twister generator [12] because of it's large key space $(2^{19937} - 1)$. Using key $K1$, a random permutation of ***blockIndices*** vector of similar size $1 \times N$ is generated. This random block vector (***scrambledIndices***) is mapped with the original ***blockIndices*** vector using one-to-one mapping strategy. Finally, all original blocks represented by ***blockIndices*** are replaced with the mapped random blocks represented by ***scrambledIndices***. *Algorithm* 1 demonstrates the complete working process of block scrambling method. The beauty of this algorithm is that the same process can be used to reverse the block scrambling of the image. Thus, the same *Algorithm* 1 will be used at the receiver side to unscramble the blocks, thereby, decrypting the image.

2) PIXEL SCRAMBLING

Once the block scrambling is done, the second layer of security, i.e., Pixel scrambling is induced to further enhance the encryption of the image. *Algorithm* 2 elaborates the

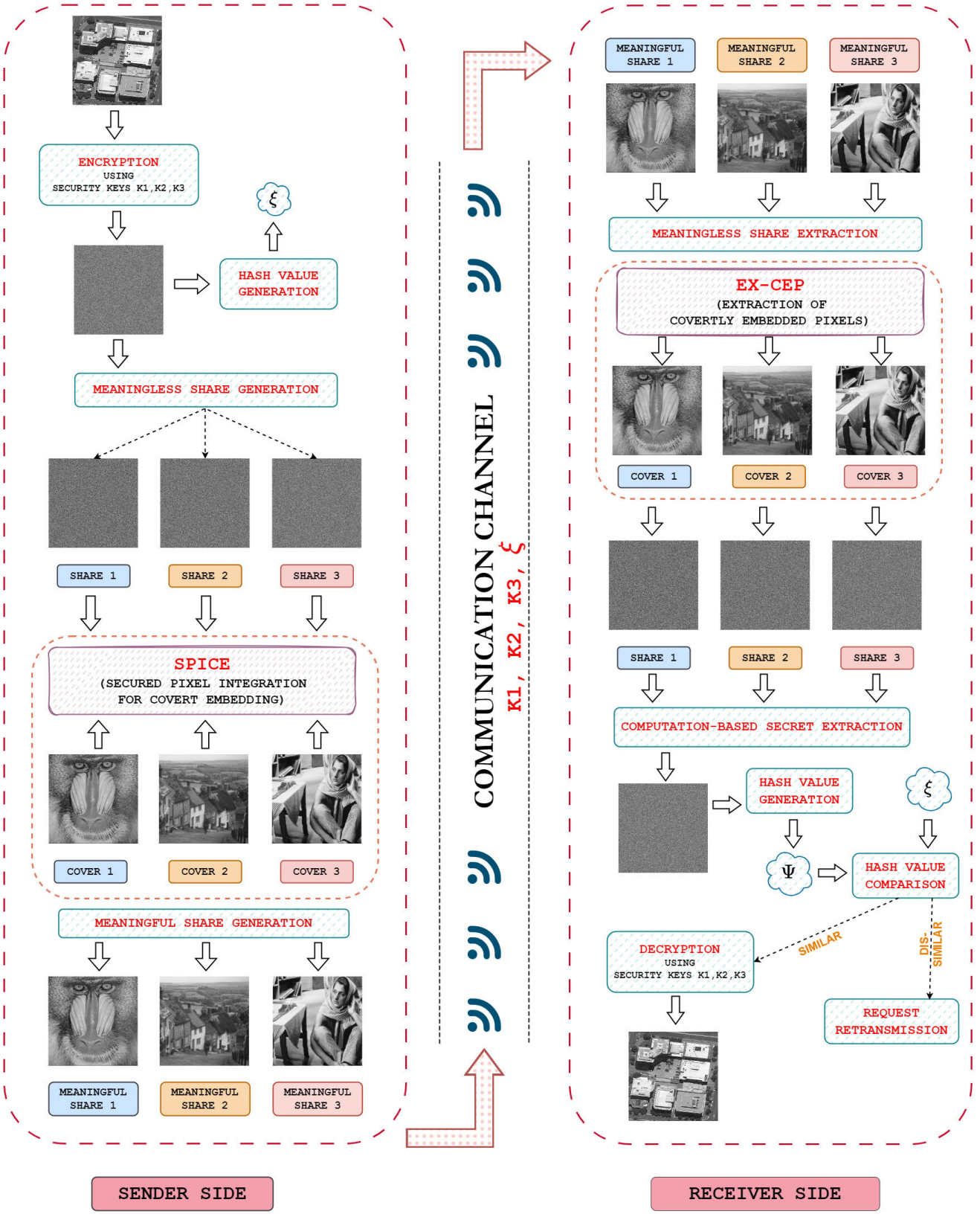

FIGURE 1. Flow chart for the proposed approach.

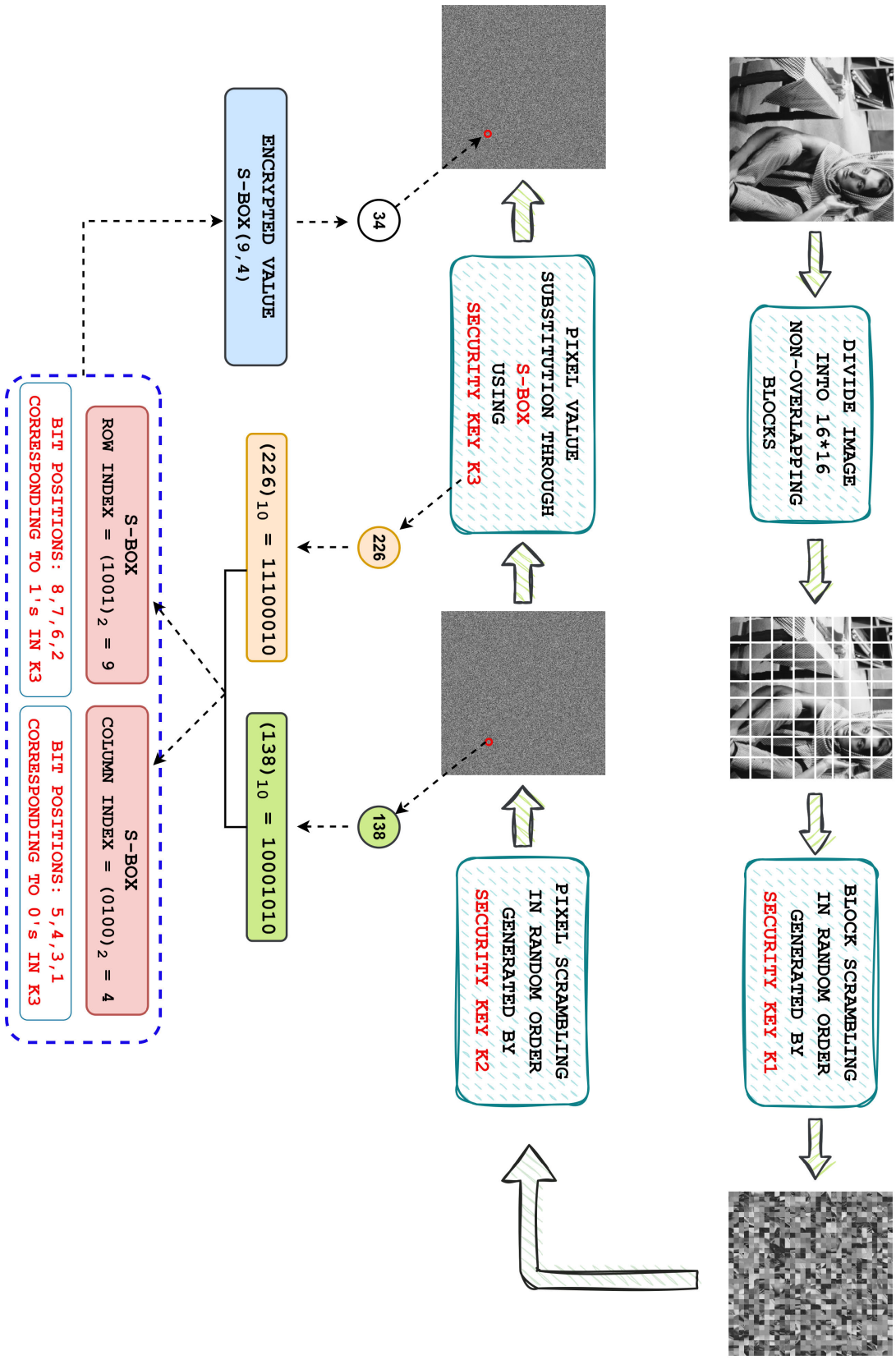

FIGURE 2. Figure showing the process of encryption adopted in our proposed approach.

Algorithm 1 Block Scrambling: Scramble Non-Overlapping Blocks Within an Image

Input
(1) **inputImage:** The input image to be scrambled.
(2) **blockSize:** The size of the non-overlapping blocks.
(3) **K1:** Security key generated using Mersenne Twister generator.

Output
(1) **Image1:** The image with non-overlapping blocks scrambled randomly.

Ensure
(1) **inputImage:** is of size 512×512.

Begin Procedure:

STEP 1: Determine the dimensions of **inputImage**:

$M \leftarrow$ Number of rows in *inputImage*
$N \leftarrow$ Number of columns in *inputImage*

STEP 2: Calculate the number of non-overlapping blocks in both dimensions:

$numBlocksX \leftarrow floor(M/blockSize)$
$numBlocksY \leftarrow floor(N/blockSize)$

STEP 3: Create an array **blockIndices** with indices from 1 to **(numBlocksX** $\times$ **numBlocksY)**.

STEP 4: Generate a random permutation **scrambledIndices** of **blockIndices** where security key $K1$ is used as the seed of the random number generator.

$scrambledIndices \leftarrow randperm(blockIndices, K1)$

STEP 5: Initialize an empty matrix **Image1** of the same size as **inputImage**.

$Image1 \leftarrow zeros(size(inputImage))$

STEP 6: Repeat *Steps* **(A)** to **(E)** for each i in 1 to $(numBlocksX \times numBlocksY)$:

(A) Calculate the source block index **SblockIndex**, source block row index **SrowIndex**, and source block column index **ScolIndex** corresponding to i.

$$SblockIndex \leftarrow scrambledIndices(i)$$
$$SrowIndex \leftarrow mod((SblockIndex - 1), numBlocksX) \times blockSize + 1$$
$$ScolIndex \leftarrow ((SblockIndex - 1)/numBlocksX) \times blockSize + 1$$

(B) Calculate the destination block index **DblockIndex**, destination block row index **DrowIndex**, and destination block column index **DcolIndex** corresponding to the current iteration index i.

$$DblockIndex \leftarrow blockIndices(i)$$
$$DrowIndex \leftarrow mod((DblockIndex - 1), numBlocksX) \times blockSize + 1$$
$$DcolIndex \leftarrow ((DblockIndex - 1)/numBlocksX) \times blockSize + 1$$

(C) Extract the source block **Sblock** from **inputImage** using **SrowIndex** and **ScolIndex**.

$$Sblock \leftarrow inputImage(SrowIndex : (SrowIndex + blockSize - 1), ScolIndex : (ScolIndex + blockSize - 1))$$

(D) Extract the destination block **Dblock** from **inputImage** using **DrowIndex** and **DcolIndex**.

$$Dblock \leftarrow inputImage(DrowIndex : (DrowIndex + blockSize - 1), DcolIndex : (DcolIndex + blockSize - 1))$$

(E) Swap **Sblock** and **Dblock** in **Image1** at their respective positions.

Place *Sblock* in $Image1$ at $(DrowIndex, DcolIndex)$
Place *Dblock* in $Image1$ at $(SrowIndex, ScolIndex)$

End of Procedure

process of Pixel scrambling. The scrambled image received from *Algorithm* 1 is given as an input to the Pixel scrambling algorithm. In the block scrambled image, the blocks as a whole are shuffled within the image. However, the pixels (location) inside the block remains intact. Therefore, in pixel scrambling, the pixels inside the 16×16 non-overlapping blocks are scrambled further, which would change their locations, thereby, ensuring enhanced randomness within the image. In order to make it more authentic, a symmetric secret key $K2$ is used which is generated by Philox generator [13]. This generator supports multiple stream and substream. The search space of this generator is 2^{193}. Using key $K2$, a random permutation of ***pixelIndices*** vector, i.e., ***randomIndices*** is generated, where ***pixelIndices*** represent a row vector of pixel indices of size 1×256 (as every block contains 256 (16×16) pixels) of the form: ***pixelIndices*** $\in \{1, 2, \ldots, 256\}$. The pixels in the block are then scrambled as per the indices mentioned in the ***randomIndices*** vector. This process is repeated for all the blocks of the secret image. The image obtained after block and pixel scrambling is passed to the final

Algorithm 2 Pixel Scrambling: Scramble Pixels Within a Block

Input

(1) **Image1:** The input image to be scrambled pixel wise.
(2) **blockSize:** The size of the non-overlapping blocks.
(3) **K2:** Security key generated using Philox generator.

Output

(1) **Image2:** Image having pixels scrambled randomly in all the blocks.

Ensure

(1) **Image1:** is of size 512×512.

Begin Procedure:

STEP 1: Determine the dimensions of **Image1**:

$M \leftarrow$ Number of rows in $Image1$
$N \leftarrow$ Number of columns in $Image1$

STEP 2: Initialize an empty matrix **Image2** of the same size as **Image1**.

$Image2 \leftarrow zeros(size(Image1))$

STEP 3: Calculate the number of non-overlapping blocks in both dimensions:

$numBlocksX \leftarrow floor(M/blockSize)$
$numBlocksY \leftarrow floor(N/blockSize)$

STEP 4: Determine the number of pixels of **inputBlock** (whose pixels we wish to scramble).

$numPixels \leftarrow blockSize \times blockSize$

STEP 5: Create an array **pixelsIndices** with indices from 1 to *numPixels*.

STEP 6: Generate a random permutation **randomIndices** of **pixelsIndices** where security key $K2$ is used as the seed of the random number generator.

$randomIndices \leftarrow randperm(pixelsIndices, K2)$

STEP 7: Repeat *Steps* **(A)** to **(G)** for each i in 1 to **(numBlocksX** $\times$ **numBlocksY)**:

(A) Determine row index **rowIndex**, and column index **colIndex** of the **inputBlock** corresponding to the current iteration index i:

$blockIndex \leftarrow i$
$rowIndex \leftarrow mod((blockIndex - 1), numBlocksX) \times blockSize + 1$
$colIndex \leftarrow ((blockIndex - 1)/numBlocksX) \times blockSize + 1$

(B) Extract the **inputBlock** from **Image1** using **rowIndex** and **colIndex**.

$inputBlock \leftarrow Image1(rowIndex : (rowIndex + blockSize - 1),$
$colIndex : (colIndex + blockSize - 1))$

(C) Reshape **inputBlock** into a $1D$ column vector:

$reshapedBlock \leftarrow reshape(inputBlock, [numPixels, 1])$

(D) Initialize an empty $1D$ column vector **scrambledBlock** of the same size as **reshapedBlock**.

$scrambledBlock \leftarrow zeros(size(reshapedBlock))$

(E) Repeat *Steps* **(i)** to **(v)** for each j in 1 to **numPixels** (of **inputBlock**):

(i) Calculate the source pixel index **sPixelIndex**.

$sPixelIndex \leftarrow randomIndices(j)$

(ii) Calculate the destination source pixel index **dPixelIndex**.

$dPixelIndex \leftarrow pixelsIndices(j)$

(iii) Extract the source pixel **sPixel** from **reshapedBlock** using **sPixelIndex**.

$sPixel \leftarrow reshapedBlock(sPixelIndex, 1)$

(iv) Extract the destination pixel **dPixel** from **reshapedBlock** using **dPixelIndex**.

$dPixel \leftarrow reshapedBlock(dPixelIndex, 1)$

(v) Swap **sPixel** and **dPixel** in **scrambledBlock** at their respective positions.

$scrambledBlock(sPixelIndex, 1) \leftarrow dPixel$
$scrambledBlock(dPixelIndex, 1) \leftarrow sPixel$

(F) Reshape **scrambledBlock** back to the original shape, i.e., $2D$ matrix of size **blockSize** $\times$ **blockSize**:

$scrambledBlock \leftarrow reshape(scrambledBlock, [blockSize, blockSize])$

(G) Place **scrambledBlock** in **Image2** at **(rowIndex, colIndex)**.

End of Procedure

Algorithm 3 S-Box Value Substitution

Input

(1) **Image2:** The input image.
(2) $\boldsymbol{\varphi}$: Matrix of size 16×16 storing the S-Box values.
(3) **K3:** Security key generated using **Threefry generator** such that their middle 8 bits have **Hamming Distance** of 4.

Output

(1) **Image3:** The image with substituted values.

Begin Procedure:

STEP 1: Determine the dimensions of **Image2**:

$M \leftarrow$ Number of rows in $Image2$
$N \leftarrow$ Number of columns in $Image2$

STEP 2: Initialize an empty matrix **Image3** of the same size as **Image2**.

$Image3 \leftarrow zeros(size(Image2))$

STEP 3: Compute the binary equivalent of **K3**, extract the middle eight bits and store them in $1D$ row vector $\boldsymbol{\zeta}$ of size 1×8.

$$\theta \leftarrow binary(K3)$$
$$\zeta \leftarrow middle8bits(\theta) \quad \text{s.t.,} \quad HAMMING(\zeta) = 4$$

STEP 4: Initialize two $1D$ row vectors $\boldsymbol{\vartheta}$ and $\boldsymbol{\eta}$ of size 1×4, which stores the bit locations of $\boldsymbol{\zeta}$ having value 0 and 1, respectively.

$$\vartheta = \left\{ k \mid \zeta(k) = 0 \quad \text{and} \quad k \in \{1, 2, 3, \ldots, 8\} \right\}$$
$$\eta = \left\{ k \mid \zeta(k) = 1 \quad \text{and} \quad k \in \{1, 2, 3, \ldots, 8\} \right\}$$

STEP 5: Repeat *Step* **(A)** for each i in 1 to $\boldsymbol{M}$:

(A) Repeat *Steps* **(i)** to **(vi)** for each j in 1 to $\boldsymbol{N}$:

(i) Let $\boldsymbol{P_c}$ be the pixel in **Image2** at row index $\boldsymbol{i}$ and column index $\boldsymbol{j}$.

$$P_c \leftarrow Image2(i, j)$$

(ii) Compute the binary equivalent of $\boldsymbol{P_c}$ and store them in $1D$ row vector $\boldsymbol{\tau}$ of size 1×8.

$$\tau \leftarrow binary(P_c)$$

(iii) Initialize two $1D$ row vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of size 1×4, which stores the bit values from $\boldsymbol{\tau}$ corresponding to locations stored in $\boldsymbol{\vartheta}$ and $\boldsymbol{\eta}$, respectively.

$$\alpha = \left\{ \tau(k) \mid k \in \{\vartheta(1),\ \vartheta(2),\ \vartheta(3),\ \vartheta(4)\} \right\}$$
$$\beta = \left\{ \tau(l) \mid l \in \{\eta(1),\ \eta(2),\ \eta(3),\ \eta(4)\} \right\}$$

(iv) Compute the row index $\boldsymbol{\upsilon}$ and column index $\boldsymbol{\omega}$ of matrix $\boldsymbol{\varphi}$ from row vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

$$\upsilon \leftarrow decimal(\alpha)$$
$$\omega \leftarrow decimal(\beta)$$

(v) Let $\boldsymbol{\Upsilon}$ be the grayscale value present at row index $\boldsymbol{\upsilon}$ and column index $\boldsymbol{\omega}$ in matrix $\boldsymbol{\varphi}$.

$$\Upsilon \leftarrow \varphi(\upsilon, \omega)$$

(vi) Substitute the value $\boldsymbol{\Upsilon}$ in **Image3** at row index $\boldsymbol{i}$ and column index $\boldsymbol{j}$.

$$Image3(i, j) \leftarrow \Upsilon$$

End of Procedure

stage of encryption process to add a third layer of encryption through S-Box value substitution method.

3) S-BOX VALUE SUBSTITUTION

This is the final stage of proposed image encryption method. In this step, a 16×16 decimal valued Rijndael S-box (substitution-box) [14] is considered for substituting the values of (block as well as pixel) scrambled image (received from the *Algorithm* 2) with those mentioned in S-box. The values inside the S-box are grayscale values ranging from $[0, \ldots, 255]$. Figure 3 shows the decimal valued S-box considered in our work. For sake of clarity, let us assume the output of *Algorithm* 2 be *Image*2. The main idea is to break down the grayscale intensity value of every pixel P_c in *Image*2 into it's binary equivalent of 8-bits. From these 8-bits, two nibbles (of 4-bits each) are formed, such that, one nibble determines the column index and other nibble determines the row-index of the value (Υ) within the S-box which would substitute the value of P_c in *Image*2. The construction of two nibbles is carried with the help of a symmetric secret key $K3$ generated using Threefry generator with key space of 2^{514}. In the proposed encryption strategy, one constraint is imposed on the generation of $K3$, i.e., the Hamming weight of the middle 8-bits of the binary form of selected $K3$ should be equal to four (4). Hamming weight of four would ensure the occurrence of four $1s$ and four $0s$ in the middle 8-bits of $K3$. The bit locations corresponding to four $1s$ and four $0s$ will help in the construction of two nibbles α and β, respectively. This can can be better understood with the help of an example shown in Figure 2. In this

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	99	124	119	123	242	107	111	197	48	1	103	43	254	215	171	118
1	202	130	201	125	250	89	71	240	173	212	162	175	156	164	114	192
2	183	253	147	38	54	63	247	204	52	165	229	241	113	216	49	21
3	4	199	35	195	24	150	5	154	7	18	128	226	235	39	178	117
4	9	131	44	26	27	110	90	160	82	59	214	179	41	227	47	132
5	83	209	0	237	32	252	177	91	106	203	190	57	74	76	88	207
6	208	239	170	251	67	77	51	133	69	249	2	127	80	60	159	168
7	81	163	64	143	146	157	56	245	188	182	218	33	16	255	243	210
8	205	12	19	236	95	151	68	23	196	167	126	61	100	93	25	115
9	96	129	79	220	34	42	144	136	70	238	184	20	222	94	11	219
10	224	50	58	10	73	6	36	92	194	211	172	98	145	149	228	121
11	231	200	55	109	141	213	78	169	108	86	244	234	101	122	174	8
12	186	120	37	46	28	166	180	198	232	221	116	31	75	189	139	138
13	112	62	181	102	72	3	246	14	97	53	87	185	134	193	29	158
14	225	248	152	17	105	217	142	148	155	30	135	233	206	85	40	223
15	140	161	137	13	191	230	66	104	65	153	45	15	176	84	187	22

FIGURE 3. Figure showing 16 × 16 S-box decimal values used in our proposed approach.

example, the middle 8-bits of secret key $K3$ is assumed to be $(226)_{10} = (11100010)_2$ with Hamming weight of four. Now the bit positions corresponding to four 1s in $(K3)_2$ are located, and stored in 1D vector η. Similarly, the bit positions corresponding to four 0s in $(K3)_2$ is stored in 1D vector ϑ. In our example, $\eta = (8, 7, 6, 2)$ and $\vartheta = (5, 4, 3, 1)$. Here, the bit position "1" corresponds to LSB and bit position "8" corresponds to MSB. Suppose the intensity value of the pixel P_c which needs to be encrypted is 138. The 8-bit binary equivalent of P_c is $(138)_{10} = (10001010)_2$. Now two 1D vectors α and β of size similar to η and ϑ are created. The vector α contains all the bit values of $(P_c)_2$ which are at bit positions mentioned in vector η. Similarly, β contains all the bit values of $(P_c)_2$ which are at bit positions mentioned in vector ϑ. The 1D vectors α and β are the two nibbles whose decimal values are treated as row and column indices of the S-Box matrix. In context to our example, the vectors $\alpha = (1001)$ and $\beta = (0100)$. Thus, the value corresponding to $rowindex = (\alpha)_{10} = 9$ and $columnindex = (\beta)_{10} = 4$ in S-box is $\Upsilon = 34$ (as can be observed from Figure 3). Finally, the Υ value (i.e., 34) will substitute the original intensity value of P_c (i.e., 138) in *Image*2. This process is repeated for all the pixels in *Image*2. The detailed process of S-Box value substitution is mentioned in *Algorithm* 3. The working of the proposed substitution method is shown in Figure 2.

B. HASH VALUE GENERATION

The proposed encryption methodology explained in Section III-A ascertains the *Confidentiality* and *Authentication* of the secret image. Now, apart from confidentiality and authentication, an another important security requirement from any effective VC scheme is *Integrity*. The proposed approach achieves *Integrity* through the use of novel hash value generation strategy, explained herewith. In the proposed work, a 128-bit hash value is generated from the encrypted secret image. Let us assume, *Image*3 to be output of *Algorithm* 3. In our work, the size of *Image*3 will be 512×512 (this is because of our assumption that the size of input secret (satellite) image is 512×512). First of all, *Image*3 is divided into non-overlapping blocks of size 16×16. Thus, for 512×512 image, there will be total of 1024 blocks (32×32). Now, from every block, one pixel is randomly selected using two secret keys χ and ψ derived from secret keys $K1$, $K2$ and $K3$. The row index of the pixel within the 16×16 block is randomly selected using derived secret key χ, where, $\chi \leftarrow K1 \oplus K2$. Similarly, the column index of pixel is randomly selected using derived secret key ψ, where, $\psi \leftarrow K2 \oplus K3$. On repeating this process for every block, a sub-image (*SubImage*) of size 32×32 is generated, constituting of 1024 pixels, one random pixel from each of the 1024 blocks of encrypted image. Subsequently, the *SubImage* is further sub-divided into non-overlapping blocks of size 8×8. Consequently, a total of 16 such blocks will be there in *SubImage*. From every 8×8 block, an 8-bit binary hash value is deduced by performing bit-wise XOR operation between all the 64 pixels of the block in zig-zag manner. The 8-bit hash pattern from each block is appended together resulting into a bit stream of $128(16 \times 8)$ bits. This 128-bit stream is the resultant hash value of the encrypted secret image. The sender shares this hash value with the receiver for integrity

verification. A similar hash value is computed by the receiver from the encrypted secret image prior to it's decryption. If any intentional or unintentional attacks happens on the secret image directly or on any of it's share, then, the received hash value and the recalculated hash value of the secret image will mismatch. The integrity of the received secret image can thus be substantiated through match/mismatch of the hash values at the receiver side. *Algorithm* 4 elucidates the complete process of hash value generation. The general outline of the process of hash value generation from the encrypted image is illustrated in Figure 4.

C. MEANINGFUL SHARE CONSTRUCTION

Once the secret image is encrypted, the next phase of the proposed approach works towards securely transmitting it over the channel. In this phase, a secret sharing approach is adopted, where three gray valued meaningful shares are generated for the encrypted secret image and transmitted over the channel via different routes. It is impossible to reveal the contents of the secret image using insufficient number of shares in case of man-in-the-middle attack. One can not reveal the contents of the secret image using one or two shares only, even in the presence of infinite computation power. To further solidify the cryptographic capability of the proposed approach, meaningless shares are converted into meaningful shares by hiding them into three grayscale natural cover images. Meaningful shares are necessary measure to avoid any cryptanalysis or man-in-the-middle passive attacks. The process of generating meaningful shares in the proposed approach is accomplished through two steps:

1) In the first step, three meaningless shares are generated using polynomial equations of degree two.
2) A novel method called SPICE (Secured Pixel Integration For Covert Embedding) is used to create three (grayscale) meaningful shares from three (grayscale) natural cover images and three (grayscale) meaningless shares.

1) MEANINGLESS SHARE GENERATION

In this section, the process of generating three meaningless shares from the encrypted secret image has been elaborated. Let us assume that the (input) encrypted secret image is denoted as *Image*3. For every $(i,j)_{th}$ pixel in *Image*3, three grayscale values in the range of $[0,\ldots,255]$ are derived corresponding to three pixels (one each) in three meaningless shares at index (i,j),i.e., $MLESS^1_{SHARE}(i,j)$, $MLESS^2_{SHARE}(i,j)$ and $MLESS^3_{SHARE}(i,j)$. Let P be the $(i,j)^{th}$ pixel of encrypted image. The computation of intensity values for $MLESS^1_{SHARE}(i,j)$, $MLESS^2_{SHARE}(i,j)$ and $MLESS^3_{SHARE}(i,j)$ from the secret pixel P is done through the use of a quadratic equation $Ax^2 + Bx + C = E(x)$, where $C = P$ and A and B are chosen randomly between -10 to 10. The output of quadratic equation, i.e., $E(x)$ is calculated for three different values of x, i.e., $E(1), E(2), E(3)$ for $x = 1, 2, 3$, respectively. If the calculated value(s) is less than 0 or greater than 255, then, the coefficients A and B are re-estimated between -10 to 10, (repeatedly,) until the values $E(1), E(2), E(3) \in \{0,\ldots,255\}$. Thereafter, the following assignments are made, thereby, generating three meaningless shares.

$$MLESS^1_{SHARE}(i,j) = E(1) \tag{1}$$
$$MLESS^2_{SHARE}(i,j) = E(2) \tag{2}$$
$$MLESS^3_{SHARE}(i,j) = E(3) \tag{3}$$

The above process is repeated for all pixels of *Image*3, $i \in \{1,\ldots,512\}$, $j \in \{1,\ldots,512\}$, where, a distinct quadratic equation is used for every pixel. Consequently, the generated three meaningless shares are of the same dimension as the encrypted secret image. *Algorithm* 6 explains the above-mentioned process of meaningless share generation in detail.

2) SPICE: SECURED PIXEL INTEGRATION FOR COVERT EMBEDDING

This section elaborates on the generation of meaningful shares from meaningless shares computed in previous section. The construction of meaningful shares employs the use of natural images which would act as a cover to conceal the meaningless shares within them. The cover images considered in our work are natural grayscale images (for example, Barbara, Baboon, etc.) which will be displayed on the meaningless shares to make them meaningful. In the proposed approach, a novel method, i.e., *SPICE* (Secured Pixel Integration For Covert Embedding) has been presented which strategically embed the meaningless shares pixels within the pixels of cover images. Figure 5 demonstrates the complete process of meaningful share generation from the corresponding meaningless shares. The step-by-step description of *SPICE* method has been detailed in *Algorithm* 7. Let us assume the three meaningless shares denoted as $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$ and $MLESS^3_{SHARE}$ are of size $M \times N$ (where $M = N = 512$ in our work).In the SPICE method, three random matrices RM^1_{SHARE}, RM^2_{SHARE}, and RM^3_{SHARE} of size $2M \times 2N$ are generated randomly consisting of integer values in the range $[0,\ldots,999]$ using three derived secret keys χ, ψ, ϕ. These keys are calculated using already known secret keys $K1$, $K2$ and $K3$ as per the following equations:

$$\chi \leftarrow (K1 \oplus K2) \,\&\&\, (\sim K3) \tag{4}$$
$$\psi \leftarrow ((K1 \,||\, K2) \oplus K3) \,\&\&\, (\sim K1) \tag{5}$$
$$\phi \leftarrow (((\sim K2) \oplus (\sim K3)) \,||\, K1) \tag{6}$$

Now, three different gray valued cover images $COVER^1$, $COVER^2$, and $COVER^3$ of size $2M \times 2N$ are selected. These cover images will act as the face of the meaningful shares. All three random matrices and three cover images are divided into non-overlapping blocks of size 2×2. The total number of such 2×2 blocks in cover image $COVER^i$ and random matrix RM^i_{SHARE} will be MN which is the same as the number of pixels in the corresponding meaningless shares. In order

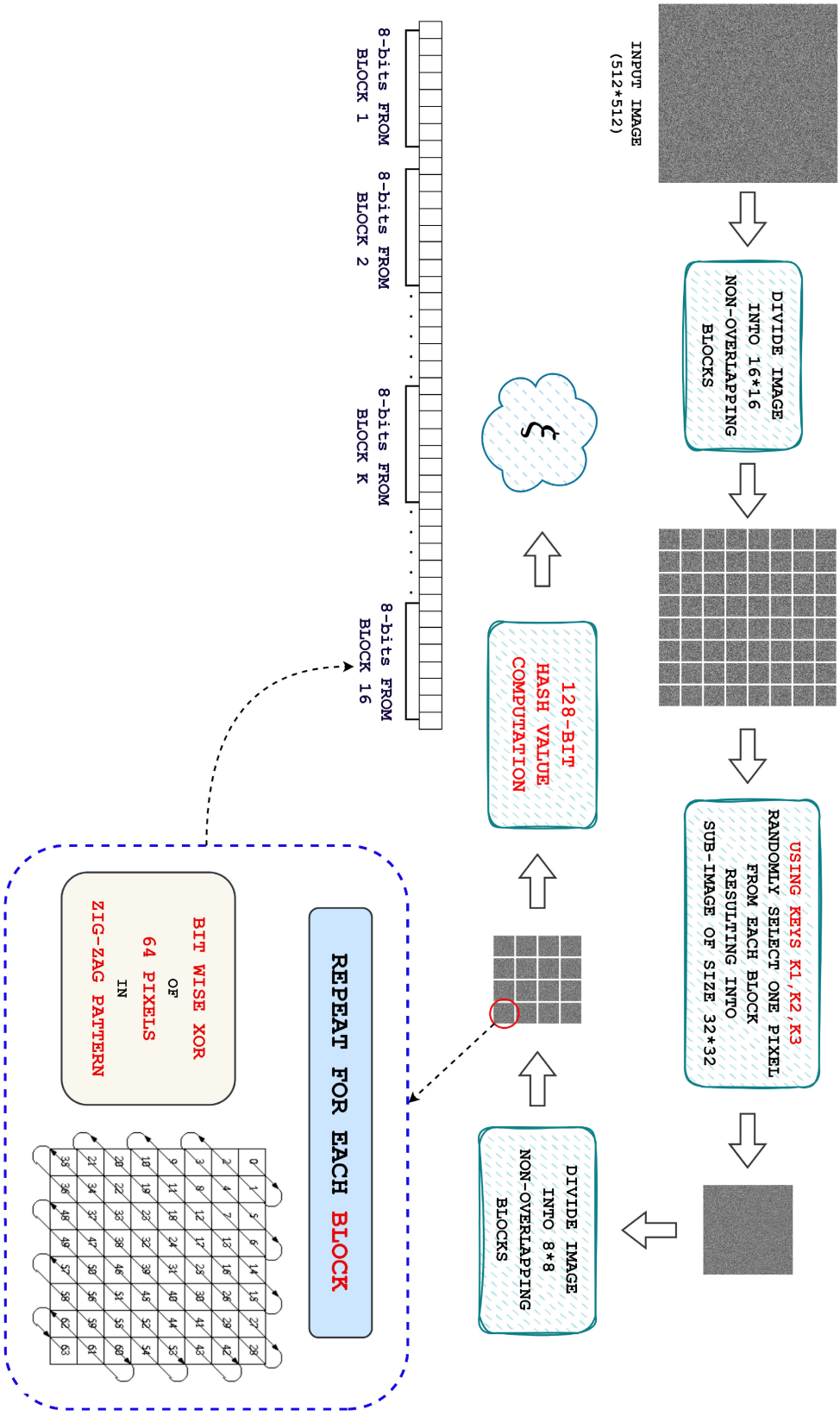

FIGURE 4. Figure showing the process of generating hash value from the encrypted image in our proposed approach.

Algorithm 4 Hash Value Generation

Input

(1) **Image3:** The input image whose hash value is to be generated.
(2) **blocksize**: The size of the non-overlapping blocks, i.e., 16×16.
(3) Security Keys **K1**, **K2**, and **K3**.

Output

(1) $\boldsymbol{\xi}$: Hash Value of **Image3**.

Ensure

(1) **Image3:** is of size 512×512.

Begin Procedure:

STEP 1: Determine the dimensions of **Image3**:

$M \leftarrow$ Number of rows in $Image3$
$N \leftarrow$ Number of columns in $Image3$

STEP 2: Calculate the number of non-overlapping blocks in both dimensions:

$numBlocksX \leftarrow 32 \leftarrow floor(M/blockSize)$
$numBlocksY \leftarrow 32 \leftarrow floor(N/blockSize)$

STEP 3: Initialize an empty matrix **SubImage** of the same size as **(numBlocksX** $\times$ **numBlocksY)**.

$SubImage \leftarrow zeros(32, 32) \leftarrow zeros(numBlocksX, numBlocksY)$

STEP 4: Generate two new keys $\boldsymbol{\chi}$ and $\boldsymbol{\psi}$ using bit-wise XOR operation between security keys **K1**, **K2**, and **K3**.

$\chi \leftarrow K1 \oplus K2$
$\psi \leftarrow K2 \oplus K3$

STEP 5: Repeat *Step* **(A)** for each i in 1 to **numBlocksX**:

(A) Repeat *Steps* **(i)** to **(iii)** for each j in 1 to **numBlocksY**:

(i) Determine the lower bound and upper bound of row index, and column index of the current block:

$blockIndex \leftarrow \big((i-1) \times numBlocksX\big) + j$
$rowIndexLB \leftarrow mod((blockIndex - 1), numBlocksX) \times blockSize + 1$
$colIndexLB \leftarrow ((blockIndex - 1)/numBlocksX) \times blockSize + 1$
$rowIndexUB \leftarrow rowIndexLB + blockSize - 1$
$colIndexUB \leftarrow colIndexLB + blockSize - 1$

(ii) Randomly select one pixel $\boldsymbol{P}$ from the **inputBlock** using keys $\boldsymbol{\chi}$ and $\boldsymbol{\psi}$:

$XIndex \leftarrow rand([rowIndexLB, rowIndexUB], \chi)$
$YIndex \leftarrow rand([colIndexLB, colIndexUB], \psi)$
$P \leftarrow Image3(XIndex, YIndex)$

(iii) Substitute the pixel value $\boldsymbol{P}$ in **SubImage**:

$SubImage(i, j) \leftarrow P$

STEP 6: Divide the **SubImage** into non-overlapping blocks of size 8×8.

STEP 7: Calculate the number of non-overlapping blocks in both dimensions in **SubImage**:

$nBLKsX \leftarrow 4 \leftarrow floor(numBlocksX/8)$
$nBLKsY \leftarrow 4 \leftarrow floor(numBlocksY/8)$

STEP 8: Initialize an empty $1D$ row vector $\boldsymbol{\xi}$ of size $1 \times (nBLKsX \times nBLKsY \times \mathbf{8})$:

$\xi \leftarrow zeros(1, 128) \leftarrow zeros(1, nBLKsX \times nBLKsY \times 8)$

STEP 9: Repeat *Steps* **(A)** to **(E)** for each k in 1 to **(nBLKsX** $\times$ **nBLKsY)**:

(A) Determine the row index **RI**, and column index **CI** of the **inputBlock** corresponding to the current iteration index $\boldsymbol{k}$:

$BI \leftarrow k$
$RI \leftarrow mod\big((BI - 1), nBLKsX\big) \times blockSize + 1$
$CI \leftarrow ((BI - 1)/nBLKsX) \times blockSize + 1$

(B) Extract the **inputBlock** from **SubImage** using **RI** and **CI**.

$inputBlock \leftarrow SubImage(RI : (RI + blockSize - 1),$
$CI : (CI + blockSize - 1))$

(C) Initialize $\mathfrak{S} \leftarrow 00000000$, $x \leftarrow 1$, $y \leftarrow 1$

(D) Repeat *Steps* **(i)** to **(iv)** for each $\boldsymbol{z}$ in 1 to 8×8:

(i) Let $\boldsymbol{P_c}$ be the pixel in **inputBlock** at row index $\boldsymbol{x}$ and column index $\boldsymbol{y}$.

$P_c \leftarrow inputBlock(x, y)$

Algorithm 5 Hash Value Generation Continued

(ii) Compute the binary equivalent of $\boldsymbol{P_c}$ and store them in $1D$ row vector $\boldsymbol{\tau}$ of size 1×8.

$$\tau \leftarrow binary(P_c)$$

(iii) Compute bit-wise XOR of $\Im$ and $\boldsymbol{\tau}$ and store the resultant back in $\Im$.

$$\Im \leftarrow \Im \oplus \tau$$

(iv) $\Im$ represents the pixel wise XOR of all the pixels in the **inputBlock**, where every pixel is extracted in a $Zig-Zag$ manner, whose pseudo code is mentioned below:

```
If mod(x + y, 2) == 0 then
    If y < 8 then
        y ← y + 1
    Else
        x ← x + 2
    EndIf
    If x > 1 then
        x ← x − 1
    EndIf
Else
    If x < 8 then
        x ← x + 1
    Else
        y ← y + 2
    EndIf
    If y > 1 then
        y ← y − 1
    EndIf
EndIf
```

(E) Append the 8-bit $\Im$ at the end of $\boldsymbol{\xi}$:

$$\xi \leftarrow \xi \mid \Im$$

End of Procedure

to generate a meaningful share $MFUL^i_{SHARE}$, a meaningless share $MLESS^i_{SHARE}$ is embedded in a cover image $COVER^i$ using a random matrix RM^i_{SHARE}, where $i \in \{1, 2, 3\}$. Let $PSeq$ be a 1D vector of size $1 \times MN$ storing the pixel indices, such that, $PSeq = [0, 1, \ldots, MN]$. Now two scrambled (random) permutations, $CSeq$ (block indices of cover image) and $RSeq$ (block indices of random matrix) of $PSeq$ are generated using security keys ζ and η, respectively, where, ζ and η are derived from the known secret keys $K1, K2$ and $K3$ as follows:

$$\zeta \leftarrow (K1 \,||K2) \oplus (K1 \,\&\&\, K2) \tag{7}$$

$$\eta \leftarrow (K2 \oplus (\sim K3)) \tag{8}$$

Let P^1_i, P^1_i and P^1_i be three pixels in the three meaningless shares $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$ and $MLESS^3_{SHARE}$, respectively, corresponding to the i^{th} index location as mentioned in $PSeq$. Secondly, let $(block^1_{cover})_i$, $(block^2_{cover})_i$, $(block^3_{cover})_i$ be the three blocks of size 2×2 in the cover images $COVER^1$, $COVER^2$, and $COVER^3$ corresponding to the i^{th} index location as mentioned in $CSeq$. Lastly, let $(block^1_{RM})_i$, $(block^2_{RM})_i$, $(block^3_{RM})_i$ be the three blocks of size 2×2 in the random matrices RM^1_{SHARE}, RM^2_{SHARE}, and RM^3_{SHARE} corresponding to the i^{th} index location as mentioned in $RSeq$. Now, the idea is to embed P^k_i in one of the four pixels of $(block^k_{cover})_i$ using the four integer values of $(block^k_{RM})_i$, where, $k \in \{1, 2, 3\}$. The selection of the embedding pixel of $(block^k_{cover})_i$ is based on the output of the $modulo4$ operation returns one of the four values $\{0, 1, 2, 3\}$ when applied on the sum of integer values of $(block^k_{RM})_i$ as shown in Figure 5. The procedure is repeated for every index $i \in \{1, \ldots, MN\}$, thereby, embedding all the pixels of the three meaningless shares. Finally, three meaningful shares $MFUL^1_{SHARE}$, $MFUL^2_{SHARE}$ and $MFUL^3_{SHARE}$ will be generated at the end of this procedure.

D. SECRET EXTRACTION PHASE

Secret extraction phase of our proposed approach is executed on the receiver side. The proposed work entails the sharing of minutest of supplementary information in the form of three security keys $K1$, $K2$, $K3$ (used in the encryption, hash generation and construction of meaningful shares) and the computed hash value by the sender side with the receiver. The receiver on receiving this supplementary information (along with the three meaningful shares) extracts the meaningless shares from the meaningful ones, which are processed further for the extraction of original (encrypted) secret image. Secret extraction phase comprises of same set of steps (or algorithms) which were executed during the embedding phase at the sender side, however, the order of execution of these steps will be in reverse, i.e., 1) $EX-CEP$: Extraction of Covertly Embedded Pixels (extraction of meaningless shares), 2) Extraction of Encrypted Secret

Algorithm 6 Meaningless Share Generation

Input

(1) **Image3:** The input encrypted secret image.

(2) ***numshares***: Number of meaningless shares formed from **Image3**

Output

(1) $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$: Three Meaningless Shares of **Image3**.

Ensure

(1) **Image3:** is of size 512×512.

(2) $\boldsymbol{numshares = 3}$

Begin Procedure:

STEP 1: Determine the dimensions of **Image3**:

$M \leftarrow$ Number of rows in $Image2$

$N \leftarrow$ Number of columns in $Image2$

STEP 2: Initialize three empty matrices $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, and $\boldsymbol{MLESS^3_{SHARE}}$ of size same as of **Image3**, i.e., $\mathbf{M \times N}$.

$MLESS^1_{SHARE} \leftarrow zeros(M, N)$

$MLESS^2_{SHARE} \leftarrow zeros(M, N)$

$MLESS^3_{SHARE} \leftarrow zeros(M, N)$

STEP 3: For each pixel of **Image3**, a $(\boldsymbol{numshares - 1})$ degree polynomial is determined as mentioned in the following pseudo code.

For $i = 1$ *to* M **do**
  For $j = 1$ *to* N **do**
    While(1)
      $data_{secret} = Image3(i, j)$
      $coeff\,(1, 1) = data_{secret}$
      $coeff\,(1, 2 : numshares) = random_{integers}([-10, 10], 1, numshares - 1)$
      For $k = numshares$ *to* 1 **do**
        $temp = power(k, [0 : numshares - 1])$
        $sum(1, k) = coeff\,. * temp$
      EndFor
      $MLESS^1_{SHARE}(i, j) = sum(1, 1)$
      $MLESS^2_{SHARE}(i, j) = sum(1, 2)$
      $MLESS^3_{SHARE}(i, j) = sum(1, 3)$
      If $((MLESS^1_{SHARE}(i, j) < 0)$ ***OR*** $(MLESS^2_{SHARE}(i, j) < 0)$ ***OR*** $(MLESS^3_{SHARE}(i, j) < 0))$ ***OR*** $((MLESS^1_{SHARE}(i, j) > 255)$ ***OR*** $(MLESS^2_{SHARE}(i, j) > 255)$ ***OR*** $(MLESS^3_{SHARE}(i, j) > 255))$ **then**
        break
      Else
        continue
      EndIf
    EndWhile
  EndFor
EndFor
End of Procedure

Image from Meaningless Shares, 3) Integrity Verification, 4) Inverse S-Box Value Substitution (for Decryption), 5) Pixel Scrambling (for Decryption) and finally, 6) Block Scrambling (for Decryption), as shown in Figure 1. In following sub-sections, all the aforementioned processes have been discussed in detail.

1) EX-CEP: EXTRACTION OF COVERTLY EMBEDDED PIXELS

The working of $EX - CEP$ method is similar to *SPICE* method. It also employs the use of three random matrices RM^1_{SHARE}, RM^2_{SHARE}, and RM^3_{SHARE} of size $2M \times 2N$ which are generated using three derived secret keys χ, ψ, ϕ from symmetric keys $K1$, $K2$ and $K3$. The inputs to $EX - CEP$ are three meaningful shares, $MFUL^1_{SHARE}$, $MFUL^2_{SHARE}$, and $MFUL^3_{SHARE}$. The objective of this method is to determine three meaningless shares, $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$, and $MLESS^3_{SHARE}$ from $MFUL^1_{SHARE}$, $MFUL^2_{SHARE}$, and $MFUL^3_{SHARE}$, with the help of RM^1_{SHARE}, RM^2_{SHARE}, and RM^3_{SHARE}. Similar to *SPICE*, three 1D vectors *PSeq*, *CSeq* and *RSeq* of size $1 \times MN$ are generated. *CSeq* and *RSeq*

Algorithm 7 SPICE: Meaningful Share Construction

Input

(1) **Image3:** The input encrypted secret image of size $\boldsymbol{M} \times \boldsymbol{N}$.
(2) $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$: Three Meaningless Shares of **Image3** of size $\boldsymbol{M} \times \boldsymbol{N}$.
(3) $\boldsymbol{COVER^1}$, $\boldsymbol{COVER^2}$, $\boldsymbol{COVER^3}$: Three Cover Images of size $\mathbf{2}\boldsymbol{M} \times \mathbf{2}\boldsymbol{N}$ for three Meaningless Shares of **Image3**.
(4) Security Keys **K1**, **K2**, and **K3**.

Output

(1) $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$: Three Meaningful Shares of **Image3** of size $\mathbf{2}\boldsymbol{M} \times \mathbf{2}\boldsymbol{N}$.

Ensure

(1) **Image3**, $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$: is of size 512×512.
(2) $\boldsymbol{numshares} = \mathbf{3}$

Begin Procedure:

STEP 1: Determine the dimensions of **Image3**:

$M \leftarrow$ Number of rows in $Image3$
$N \leftarrow$ Number of columns in $Image3$

STEP 2: Initialize three empty matrices $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, and $\boldsymbol{MFUL^3_{SHARE}}$ of size $\mathbf{2}\boldsymbol{M} \times \mathbf{2}\boldsymbol{N}$.

$MFUL^1_{SHARE} \leftarrow zeros(2M, 2N)$
$MFUL^2_{SHARE} \leftarrow zeros(2M, 2N)$
$MFUL^3_{SHARE} \leftarrow zeros(2M, 2N)$

STEP 3: Generate three new keys $\boldsymbol{\phi}$, $\boldsymbol{\chi}$ and $\boldsymbol{\psi}$ using bit-wise $\boldsymbol{XOR}(\oplus)$, $\boldsymbol{OR}(||)$, $\boldsymbol{AND}(\&\&)$, $\boldsymbol{NOT}(\sim)$ operations between security keys **K1**, **K2**, and **K3**.

$\chi \leftarrow (K1 \oplus K2) \,\&\&\, (\sim K3)$
$\psi \leftarrow ((K1 \,||\, K2) \oplus K3) \,\&\&\, (\sim K1)$
$\phi \leftarrow (((\sim K2) \oplus (\sim K3)) \,||\, K1)$

STEP 4: Initialize three random matrices $\boldsymbol{RM^1_{SHARE}}$, $\boldsymbol{RM^2_{SHARE}}$, and $\boldsymbol{RM^3_{SHARE}}$ of size $\mathbf{2}\boldsymbol{M} \times \mathbf{2}\boldsymbol{N}$.

$RM^1_{SHARE} \leftarrow random_{integers}(999, 2M, 2N, \chi)$
$RM^2_{SHARE} \leftarrow random_{integers}(999, 2M, 2N, \psi)$
$RM^3_{SHARE} \leftarrow random_{integers}(999, 2M, 2N, \phi)$

STEP 5: Generate a new key $\boldsymbol{\zeta}$ using bit-wise $\boldsymbol{XOR}(\oplus)$, $\boldsymbol{OR}(||)$, $\boldsymbol{AND}(\&\&)$, $\boldsymbol{NOT}(\sim)$ operations between security keys **K1**, and **K2**.

$\zeta \leftarrow (K1 \,||K2) \oplus (K1 \,\&\&\, K2)$

STEP 6: Generate a new key $\boldsymbol{\eta}$ using bit-wise $\boldsymbol{XOR}(\oplus)$, $\boldsymbol{OR}(||)$, $\boldsymbol{AND}(\&\&)$, $\boldsymbol{NOT}(\sim)$ operations between security keys **K2**, and **K3**.

$\eta \leftarrow (K2 \oplus (\sim K3))$

STEP 7: Create a 1D sequence of pixel indices ***PSeq*** with indices from 1 to $\boldsymbol{M} \times \boldsymbol{N}$.

$PSeq = \{1, 2, 3, .., M \times N\}$

STEP 8: Generate two scrambled (random) permutations ***CSeq*** and ***RSeq*** of ***PSeq*** using security keys $\boldsymbol{\zeta}$ and $\boldsymbol{\eta}$, respectively.

$CSeq \leftarrow randperm(PSeq, \zeta)$
$RSeq \leftarrow randperm(PSeq, \eta)$

STEP 9: Divide three random matrices $\boldsymbol{RM^1_{SHARE}}$, $\boldsymbol{RM^2_{SHARE}}$, and $\boldsymbol{RM^3_{SHARE}}$ and three cover images $\boldsymbol{COVER^1}$, $\boldsymbol{COVER^2}$, $\boldsymbol{COVER^3}$ into non-overlapping blocks of size $\mathbf{2} \times \mathbf{2}$. The count of blocks is $\boldsymbol{MN}$ similar to number of pixels in **Image3**. One pixel of **Image3** will be embedded in one block.

STEP 10: Repeat *Step* **(A)** for each i in 1 to M:

(A) Repeat *Steps* **(i)** to **(iv)** for each j in 1 to N:

(i) Determine the 2×2 block of random matrices $\boldsymbol{RM^1_{SHARE}}$, $\boldsymbol{RM^2_{SHARE}}$, $\boldsymbol{RM^3_{SHARE}}$ and cover images $\boldsymbol{COVER^1}$, $\boldsymbol{COVER^2}$, $\boldsymbol{COVER^3}$ where the pixels of $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$ will be embedded.

$P^1 \leftarrow MLESS^1_{SHARE}(i, j)$ $\quad P^2 \leftarrow MLESS^2_{SHARE}(i, j)$ $\quad P^3 \leftarrow MLESS^3_{SHARE}(i, j)$
$P_{index} \leftarrow ((i-1) \times M) + j$
$blockindex_{cover} \leftarrow CSeq(P_{index})$
$blockindex_{RM} \leftarrow RSeq(P_{index})$
$rowIndexLB_{cover} \leftarrow mod((blockindex_{cover} - 1), M) \times 2 + 1$
$colIndexLB_{cover} \leftarrow ((blockindex_{cover} - 1)/M) \times 2 + 1$
$rowIndexUB_{cover} \leftarrow rowIndexLB_{cover} + 1$

Algorithm 8 SPICE: Meaningful Share Construction Continued

$$colIndexUB_{cover} \leftarrow colIndexLB_{cover} + 1$$
$$rowIndexLB_{RM} \leftarrow mod((blockindex_{RM} - 1), M) \times 2 + 1$$
$$colIndexLB_{RM} \leftarrow ((blockindex_{RM} - 1)/M) \times 2 + 1$$
$$rowIndexUB_{RM} \leftarrow rowIndexLB_{RM} + 1$$
$$colIndexUB_{RM} \leftarrow colIndexLB_{RM} + 1$$
$$block^1_{cover} \leftarrow COVER^1\left(rowIndexLB_{cover} : rowIndexUB_{cover}, colIndexLB_{cover} : colIndexUB_{cover}\right)$$
$$block^2_{cover} \leftarrow COVER^2\left(rowIndexLB_{cover} : rowIndexUB_{cover}, colIndexLB_{cover} : colIndexUB_{cover}\right)$$
$$block^3_{cover} \leftarrow COVER^3\left(rowIndexLB_{cover} : rowIndexUB_{cover}, colIndexLB_{cover} : colIndexUB_{cover}\right)$$
$$block^1_{RM} \leftarrow RM^1_{SHARE}\left(rowIndexLB_{RM} : rowIndexUB_{RM}, colIndexLB_{RM} : colIndexUB_{RM}\right)$$
$$block^2_{RM} \leftarrow RM^2_{SHARE}\left(rowIndexLB_{RM} : rowIndexUB_{RM}, colIndexLB_{RM} : colIndexUB_{RM}\right)$$
$$block^3_{RM} \leftarrow RM^3_{SHARE}\left(rowIndexLB_{RM} : rowIndexUB_{RM}, colIndexLB_{RM} : colIndexUB_{RM}\right)$$

(ii) Determine the location (reference shown at the end of the Algorithm) in $\boldsymbol{block^1_{cover}}, \boldsymbol{block^2_{cover}}, \boldsymbol{block^3_{cover}}$ where pixels $\boldsymbol{P^1}$, $\boldsymbol{P^2}, \boldsymbol{P^3}$ will be embedded based on the outcome of operation performed on blocks $\boldsymbol{block^1_{RM}}, \boldsymbol{block^2_{RM}}, \boldsymbol{block^3_{RM}}$, respectively.

$$sum^1 = block^1_{RM}(1,1) + block^1_{RM}(1,2) + block^1_{RM}(2,1) + block^1_{RM}(2,2)$$
$$sum^2 = block^2_{RM}(1,1) + block^2_{RM}(1,2) + block^2_{RM}(2,1) + block^2_{RM}(2,2)$$
$$sum^3 = block^3_{RM}(1,1) + block^3_{RM}(1,2) + block^3_{RM}(2,1) + block^3_{RM}(2,2)$$
$$LOC^1_{COVER} \leftarrow mod(sum^1, 4)$$
$$LOC^2_{COVER} \leftarrow mod(sum^2, 4)$$
$$LOC^3_{COVER} \leftarrow mod(sum^3, 4)$$

(iii) Substitute the pixels $\boldsymbol{P^1}, \boldsymbol{P^2}, \boldsymbol{P^3}$ $\boldsymbol{P}$ at locations $\boldsymbol{LOC^1_{COVER}}, \boldsymbol{LOC^2_{COVER}}, \boldsymbol{LOC^3_{COVER}}$ in $\boldsymbol{block^1_{cover}}, \boldsymbol{block^2_{cover}}, \boldsymbol{block^3_{cover}}$, respectively.

$$block^1_{cover}\left(LOC^1_{COVER}\right) \leftarrow P^1$$
$$block^2_{cover}\left(LOC^2_{COVER}\right) \leftarrow P^2$$
$$block^3_{cover}\left(LOC^3_{COVER}\right) \leftarrow P^3$$

(iv) Substitute the modified cover blocks, i.e., $\boldsymbol{block^1_{cover}}$, $\boldsymbol{block^2_{cover}}$, $\boldsymbol{block^3_{cover}}$, in $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$, respectively.

$$MFUL^1_{SHARE}\left(rowIndexLB_{cover} : rowIndexUB_{cover}, colIndexLB_{cover} : colIndexUB_{cover}\right) \leftarrow block^1_{cover}$$
$$MFUL^2_{SHARE}\left(rowIndexLB_{cover} : rowIndexUB_{cover}, colIndexLB_{cover} : colIndexUB_{cover}\right) \leftarrow block^2_{cover}$$
$$MFUL^3_{SHARE}\left(rowIndexLB_{cover} : rowIndexUB_{cover}, colIndexLB_{cover} : colIndexUB_{cover}\right) \leftarrow block^3_{cover}$$

End of Procedure

$$LOC_{block} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{3} & \mathbf{2} \end{bmatrix}$$

are two scrambled (random) permutations of *PSeq* which are generated using security keys ζ and η, respectively, where ζ and η are security keys derived from keys $K1$, $K2$ and $K3$. Let $(block^1_{MFUL})_i$, $(block^2_{MFUL})_i$, $(block^3_{MFUL})_i$ be three blocks of size 2×2 in the three meaningful shares, $MFUL^1_{SHARE}$, $MFUL^2_{SHARE}$, and $MFUL^3_{SHARE}$, corresponding to the i^{th} index location in *CSeq*. Similarly, let $(block^1_{RM})_i$, $(block^2_{RM})_i$, $(block^3_{RM})_i$ be three blocks of size 2×2 in the random matrices RM^1_{SHARE}, RM^2_{SHARE}, and RM^3_{SHARE} corresponding to the i^{th} index location in *RSeq*. Let P^1_i, P^1_i and P^1_i be three pixels in the three meaningless shares $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$ and $MLESS^3_{SHARE}$, respectively, which we wish to extract from $(block^1_{MFUL})_i$, $(block^2_{MFUL})_i$, $(block^3_{MFUL})_i$ with the help of four integer values in $(block^1_{RM})_i$, $(block^2_{RM})_i$, $(block^3_{RM})_i$, corresponding to the i^{th} index location in *PSeq*. Now, the identification of the meaningful pixel among the four pixels of $(block^k_{MFUL})_i$ which holds the pixel P^k_i of the meaningless share $MLESS^k_{SHARE}$ is based on the output of the *modulo*4 operation applied on the sum of integer values of $(block^k_{RM})_i$ (where, $k \in \{1, 2, 3\}$). The *modulo*4 operation returns one of the four values $\{0, 1, 2, 3\}$, which signifies the embedding location of P^k_i. This process is repeated for every $(block^k_{MFUL})_i$ of $MFUL^k_{SHARE}$, thereby, resulting into three meaningless shares $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$, and $MLESS^3_{SHARE}$. The step-by-step description of *SPICE* method has been detailed in *Algorithm* 9.

2) EXTRACTION OF SECRET IMAGE FROM MEANINGLESS SHARES

After the execution of *Algorithm* 9, the receiver will be able to extract three meaningless shares $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$ and $MLESS^3_{SHARE}$ from three meaningful shares $MFUL^1_{SHARE}$, $MFUL^2_{SHARE}$ and $MFUL^3_{SHARE}$, respectively. Now, the last step of the ladder, i.e., the retrieval of (original) secret image from the three available meaningless shares, has been explained in this section. As mentioned in Section III-C1, the sender used a polynomial function to generate three meaningless shares from the encrypted secret image (*Image*3). For every $(i, j)^{th}$ pixel of *Image*3, three gray values corresponding to a pixel in each of the

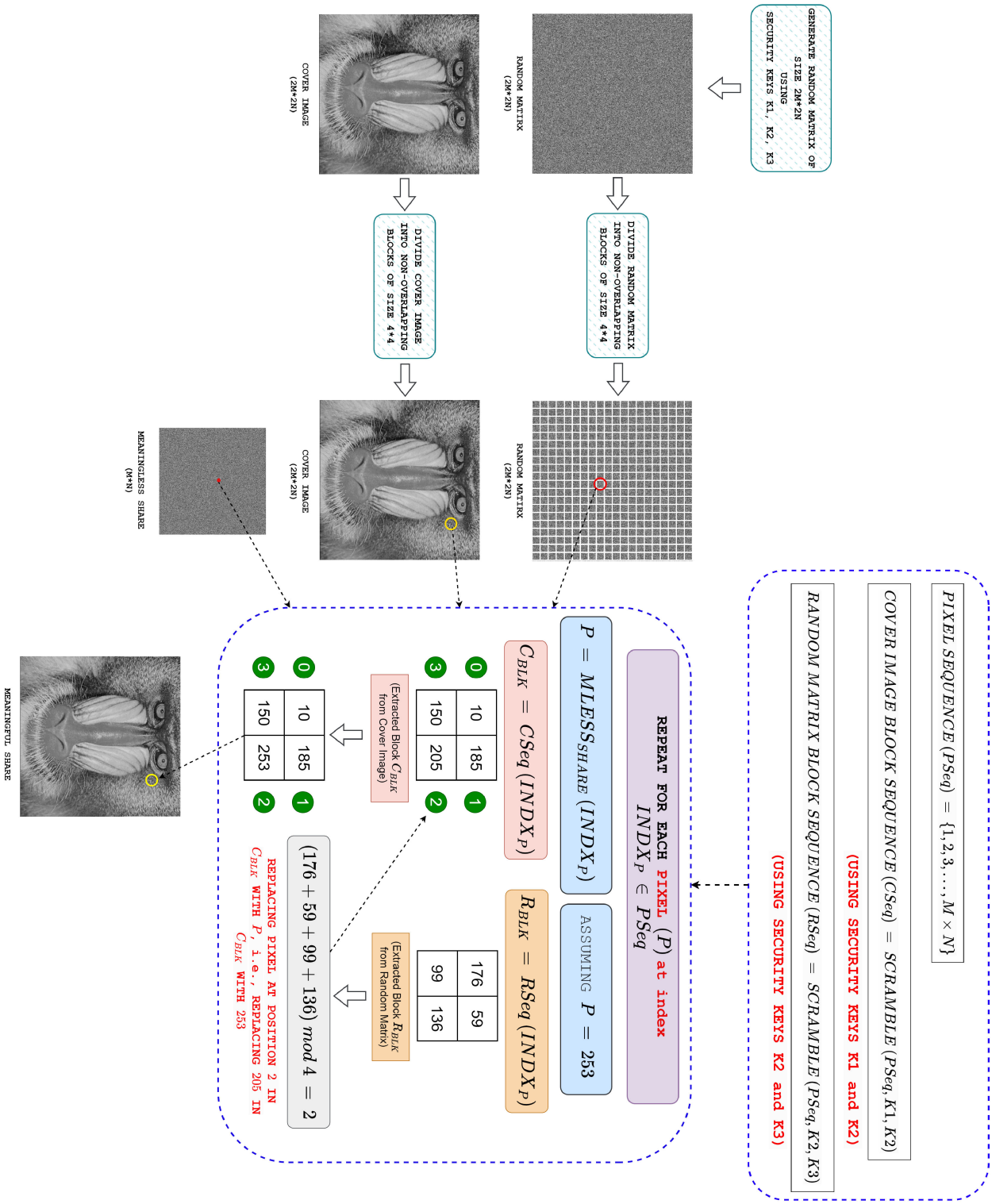

FIGURE 5. Figure showing the process of SPICE in our proposed approach.

three meaningless shares at $(i, j)^{th}$ location are generated through a distinct polynomial function. At the receiver side, the knowledge of these distinct polynomials is required to retrieve the pixel value of the (original) encrypted secret image from pixels of three meaningless shares. The proposed VC scheme employs the use of Newton polynomial basis

Algorithm 9 EX-CEP: Extraction of Covertly Embedded Pixels of Meaningless Shares From Meaningful Shares

Input

(1) $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$: Three Meaningful Shares.
(2) Security Keys **K1**, **K2**, and **K3**.

Output

(1) $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$: Three Meaningless Shares.

Ensure

(1) $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$: are of size 1024×1024.
(2) $\boldsymbol{numshares = 3}$

Begin Procedure:

STEP 1: Determine the dimensions of meaningful shares:

$MM \leftarrow$ Number of rows in $MFUL^1_{SHARE}$
$NN \leftarrow$ Number of columns in $MFUL^1_{SHARE}$

STEP 2: Initialize three empty matrices corresponding to three meaningless shares, i.e., $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, and $\boldsymbol{MLESS^3_{SHARE}}$ of size $\boldsymbol{M \times N}$, where, $M \leftarrow \frac{MM}{2}$ and $N \leftarrow \frac{NN}{2}$.

$MLESS^1_{SHARE} \leftarrow zeros(M, N)$
$MLESS^2_{SHARE} \leftarrow zeros(M, N)$
$MLESS^3_{SHARE} \leftarrow zeros(M, N)$

STEPS 3-8: Repeat STEPS 3-8 of ***Algorithm 7***: ***SPICE***.

STEP 9: Divide three random matrices $\boldsymbol{RM^1_{SHARE}}$, $\boldsymbol{RM^2_{SHARE}}$, and $\boldsymbol{RM^3_{SHARE}}$ and three meaningful shares $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$ into non-overlapping blocks $\boldsymbol{block^1_{RM}}$, $\boldsymbol{block^2_{RM}}$, $\boldsymbol{block^3_{RM}}$ and $\boldsymbol{block^1_{MFUL}}$, $\boldsymbol{block^2_{MFUL}}$, $\boldsymbol{block^3_{MFUL}}$, respectively, of size $\boldsymbol{2 \times 2}$. The count of blocks is $\boldsymbol{MN}$ similar to number of pixels in meaningless shares, $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, and $\boldsymbol{MLESS^3_{SHARE}}$. One pixel of $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, and $\boldsymbol{MLESS^3_{SHARE}}$ will be extracted from each block of $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$.

STEP 10: Repeat *Step* **(A)** for each i in 1 to M:

(A) Repeat *Steps* **(i)** to **(iii)** for each j in 1 to N:

(i) Determine the 2×2 block of random matrices $\boldsymbol{RM^1_{SHARE}}$, $\boldsymbol{RM^2_{SHARE}}$, $\boldsymbol{RM^3_{SHARE}}$ and meaningful shares $\boldsymbol{MFUL^1_{SHARE}}$, $\boldsymbol{MFUL^2_{SHARE}}$, $\boldsymbol{MFUL^3_{SHARE}}$ where the pixels of $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$ are extracted similar to STEP 10 of ***Algorithm 7***: ***SPICE***.

(ii) Determine the location (reference shown at the end of ***Algorithm 7 − 8***: ***SPICE***) in $\boldsymbol{block^1_{MFUL}}$, $\boldsymbol{block^2_{MFUL}}$, $\boldsymbol{block^3_{MFUL}}$ where pixels $\boldsymbol{P^1}$, $\boldsymbol{P^2}$, $\boldsymbol{P^3}$ corresponding to $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$, respectively, will be extracted based on the outcome of operation performed on blocks $\boldsymbol{block^1_{RM}}$, $\boldsymbol{block^2_{RM}}$, $\boldsymbol{block^3_{RM}}$, respectively.

(iii) Substitute the pixels $\boldsymbol{P^1}$, $\boldsymbol{P^2}$, $\boldsymbol{P^3}$ $\boldsymbol{P}$ at location (i, j) in $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$, respectively.

$MLESS^1_{SHARE}(i, j) \leftarrow P^1$
$MLESS^2_{SHARE}(i, j) \leftarrow P^2$
$MLESS^3_{SHARE}(i, j) \leftarrow P^3$

End of Procedure

functions to estimate them. Newton polynomial basis functions (or Newton interpolation polynomial or simply Newton polynomial) is a mathematical technique used to approximate a function with a polynomial curve. This method employs divided differences to construct a polynomial that passes through a set of given data points. The Newton polynomial is sometimes called Newton's divided differences interpolation polynomial because the coefficients of the polynomial are calculated using Newton's divided differences method.

Given a set of $n+1$ data points, i.e., $\{(x_0, y_0), (x_1, y_1), \ldots, (x_k, y_k), \ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)\}$, where no two x_k are same, the Newton polynomial $N(x)$ is given by the linear combination of Newton basis polynomial functions:

$$N(x) = \sum_{k=0}^{n} c_k n_k(x) \tag{9}$$

where, Newton basis polynomials $(n_k(x))$ are defined as:

$$n_k(x) = \prod_{i=0}^{k-1} (x - x_i) \tag{10}$$

for $k > 0$ and $n_k(x) = 1$. The coefficients c_k are defined as;

$$c_k := [y_0, \ldots, y_k] \tag{11}$$

Algorithm 10 Extraction of Secret Image From Meaningless Shares

Input

(1) $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$: Three Meaningless Shares.

Output

(1) $\boldsymbol{SECRET_{ENCRYPTED}}$: Secret Image in Encrypted Form.

Ensure

(1) $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$: are of size 512×512.

(2) $\boldsymbol{numshares = 3}$

Begin Procedure:

STEP 1: Determine the dimensions of meaningless shares:

$M \leftarrow$ Number of rows in $MLESS^1_{SHARE}$

$N \leftarrow$ Number of columns in $MLESS^1_{SHARE}$

STEP 2: Initialize an empty matrix $\boldsymbol{SECRET_{ENCRYPTED}}$ of size $\boldsymbol{M \times N}$.

$SECRET_{ENCRYPTED} \leftarrow zeros(M, N)$

STEP 3: Repeat *Step* **(A)** for each i in 1 to M:

(A) Repeat *Steps* **(i)** to **(iv)** for each j in 1 to N:

(i) For every pixel of secret image, estimate the polynomial $\boldsymbol{N(x)}$ of degree $\boldsymbol{(numshares - 1)}$ such that the constant term of the polynomial is the secret data. Instead of finding the complete polynomial, we only need to determine the constant term. This constant term can be found by using pixels, $\boldsymbol{P^1}$, $\boldsymbol{P^2}$, $\boldsymbol{P^3}$ (generated from this polynomial) corresponding to $\boldsymbol{MLESS^1_{SHARE}}$, $\boldsymbol{MLESS^2_{SHARE}}$, $\boldsymbol{MLESS^3_{SHARE}}$, respectively, by using Newton's Basis Polynomial.

$P^1 \leftarrow MLESS^1_{SHARE}(i, j)$

$P^2 \leftarrow MLESS^3_{SHARE}(i, j)$

$P^3 \leftarrow MLESS^3_{SHARE}(i, j)$

(ii) Corresponding to three meaningless shares, we have three data points, $\boldsymbol{(x_1, y_1) = (1, P^1)}$, $\boldsymbol{(x_2, y_2) = (2, P^2)}$, and $\boldsymbol{(x_3, y_3) = (3, P^3)}$, from which Newton polynomial $\boldsymbol{N(x)}$ is estimated as mentioned below:

$$N(x) \leftarrow \left[\left(\frac{y_3 - y_2}{x_3 - x_2}\right) - \left(\frac{y_2 - y_1}{x_2 - x_1}\right)\right]\left[\frac{(x - x_2)(x - x_1)}{x_3 - x_1}\right] + \left[\frac{(y_2 - y_1)(x - x_1)}{x_2 - x_1}\right] + y_1$$

(iii) The objective is to compute the value of $\boldsymbol{N(0)}$ which is the required value of pixel $\boldsymbol{P}$ of $\boldsymbol{SECRET_{ENCRYPTED}}$. The value $\boldsymbol{N(0)}$ is computed by substituting, $\boldsymbol{x = 0, x_3 = 3, x_2 = 2, x_1 = 1}$ in $\boldsymbol{N(x)}$.

$P \leftarrow N(0) \leftarrow y_3 - 3y_2 + 3y_1$

$P \leftarrow P^3 - 3P^2 + 3P^1$

(iv) Finally, substitute the value of $\boldsymbol{P}$ at location $\boldsymbol{(i, j)}$ in $\boldsymbol{SECRET_{ENCRYPTED}}$.

$SECRET_{ENCRYPTED}(i, j) \leftarrow P$

End of Procedure

where, $[y_0, \ldots, y_k]$ is the notion for (forward) divided differences defined as:

$$[y_k, \ldots, y_{k+j}] = \frac{[y_{k+1}, \ldots, y_{k+j}] - [y_k, \ldots, y_{k+j-1}]}{x_{k+j} - x_k}, \tag{12}$$

where,

$$k \in \{0, \ldots, n - j\}, \qquad j \in \{1, \ldots, n\}$$

Also,

$$[y_k] = y_k, \qquad k \in \{0, \ldots, n\} \tag{13}$$

From the above, following can be deduced:

$$[y_0] = y_0 \tag{14}$$

$$[y_0, y_1] = \frac{(y_1 - y_0)}{(x_1 - x_0)} \tag{15}$$

$$[y_0, y_1, y_2] = \frac{[y_1, y_2] - [y_0, y_1]}{(x_2 - x_0)}$$

$$[y_0, y_1, y_2] = \frac{\frac{(y_2 - y_1)}{(x_2 - x_1)} - \frac{(y_1 - y_0)}{(x_1 - x_0)}}{(x_2 - x_0)}$$

$$[y_0, y_1, y_2] = \frac{(y_2 - y_1)}{(x_2 - x_1)(x_2 - x_0)} - \frac{(y_1 - y_0)}{(x_1 - x_0)(x_2 - x_0)} \tag{16}$$

$$[y_0, y_1, y_2, y_3] = \frac{[y_1, y_2, y_3] - [y_0, y_1, y_2]}{(x_3 - x_0)} \quad (17)$$

Summarizing the above equations, the Newton polynomial can be written as:

$$N(x) = [y_0] + [y_0, y_1](x - x_0) + \ldots + [y_0, \ldots, y_n](x - x_0)(x - x_1)\ldots(x - x_{n-1}) \quad (18)$$

In our scenario, corresponding to three meaningless shares $MLESS^1_{SHARE}$, $MLESS^2_{SHARE}$ and $MLESS^3_{SHARE}$, we have three data points, $(x_1, y_1) = (1, P^1)$, $(x_2, y_2) = (2, P^2)$, and $(x_3, y_3) = (3, P^3)$, where,

$$\begin{aligned} P^1 &= MLESS^1_{SHARE}(i, j) \\ P^2 &= MLESS^2_{SHARE}(i, j) \\ P^3 &= MLESS^3_{SHARE}(i, j) \end{aligned} \quad (19)$$

For every $(i, j)^{th}$ pixel of the secret image, estimate the polynomial $N(x)$ of degree $2(i.e., numshares - 1)$ such that the constant term of the polynomial is the secret data. Instead of finding the complete polynomial, we only need to determine the constant term. This constant term can be obtained from pixels, P^1, P^2, P^3 (generated from this polynomial) using Newton polynomial basis functions. The following equation illustrates the estimation of the Newton polynomial $N(x)$:

$$N(x) \leftarrow \left[\left(\frac{y_3 - y_2}{x_3 - x_2}\right) - \left(\frac{y_2 - y_1}{x_2 - x_1}\right)\right]\left[\frac{(x - x_2)(x - x_1)}{x_3 - x_1}\right] + \left[\frac{(y_2 - y_1)(x - x_1)}{x_2 - x_1}\right] + y_1 \quad (20)$$

The objective is to compute the value of $N(0)$ which is the required value of $(i, j)^{th}$ pixel of the secret image. The value $N(0)$ is computed by substituting, $x = 0, x_3 = 3, x_2 = 2$, and $x_1 = 1$ in $N(x)$, as shown below:

$$N(0) = y_3 - 3y_2 + 3y_1 \quad (21)$$

The complete steps detailing the extraction of the secret image from three meaningless images have been mentioned in Algorithm 10.

3) INTEGRITY VERIFICATION

On successful extraction of the (encrypted) secret image from the meaningless shares by the receiver, it is imperative to verify their authenticity and integrity before their utilization for any legitimate purpose. In this process, the receiver will compute a 128-bit hash value from the extracted (encrypted) secret image using *Algorithm* 4. If any intentional or unintentional attacks happens on the secret image directly or on any of it's share, then, the received hash value and the recalculated hash value of the secret image will mismatch. The integrity of the received secret image can thus be substantiated through match/mismatch of the hash values at the receiver side. If there is a mismatch, then the receiver will request for re-transmission of shares to the sender side without forwarding it for further processing.

4) SECRET IMAGE DECRYPTION

After the integrity of the encrypted secret image has been successfully verified, it is finally, decrypted to get the original secret image. The decryption process is carried out in three stages. Firstly, the intensity values in the encrypted image are substituted with the gray values of inverse S-box shown in Figure 6 using the *Algorithm* 3. Afterwards, the substituted image is passed through *Algorithm* 2 and *Algorithm* 1 to inverse the pixel scrambling and block scrambling, respectively, done at the time of encryption. The output image obtained after the unscrambling is the original secret (satellite) image.

IV. EXPERIMENTAL RESULTS

This section presents several experiments conducted to investigate the effectiveness of our proposed VC approach in terms of imperceptibility and integrity verification as well as to compare it's performance with the existing state-of-the-art VC approaches. The proposed approach has been implemented on a computing framework comprising of Windows 11 (64 bit), MATLAB 2022b, AMD Ryzen^{TM} Threadripper^{TM} PRO 5995WX processor (2.70GHz, up to 4.5GHz Max Boost), 64GB RAM and equipped with NVIDIA® RTX^{TM} A4500 20GB. Experiments have been performed on various satellite images taken from publicly available dataset MLRSNet: A Multi-label High Spatial Resolution Remote Sensing Dataset for Semantic Scene Understanding. The dataset can be assessed at https://github.com/cugbrs/MLRSNet. In our experiments, we have considered different types of satellite images relating to the practical applications of remote sensing. Our test satellite images consist of military aircraft, freeways, forestry, parking lots, dense residential area, commercial area, industrial areas, road intersections, etc. The original MLRSNet dataset comprises of very large number of satellite images. In our experimental setup, we have considered ten randomly selected satellite images (each from different category). These ten randomly selected test images are shown in Figure 7. The efficacy of the proposed approach is demonstrated through empirical results obtained using various objective evaluation metrics (such as PSNR, MSE, SSIM, VIF, FSIM, SVDQA, PLCC, KROCC) as well as through visual results in the form of output images as meaningless and meaningful shares. The description of these objective evaluation metrics along with their ideal values has been tabulated in Table 1. Table 2 enumerates the values of different objective parameters obtained while comparing the similarity between the original and recovered secret (satellite) images using our proposed approach. Here, one can observe that the proposed approach has given ideal values of the objective parameters value which substantiates it's 100%reconstruction accuracy without any expansion or loss in contrast or information.

Figure 8 shows the final recovered secret (satellite) images corresponding to ten test images along with all the intermediate outcomes in the form of visual results. One can surmise

TABLE 1. Various objective evaluation metrics used in the experimental setup.

Objective Evaluation parameters	Procedure to calculate	Ideal value
MSE	$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j) - K(i,j))^2$	0
PSNR	$PSNR = 10log_{10}\frac{(MAX)^2}{MSE}$	∞
SSIM	$SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_1)}$ where μ_x, μ_y, σ_x^2, σ_y^2 and σ_{xy} are the average, variance and covariance for x and y respectively.	1
VIF	Visual Information fidelity is an objective image quality assessment measure that quantifies the information that is present in the reference image and how much of this reference information can be extracted from the distorted image. It is consistent with human visual system (HVS),i.e., quantifies the information that could ideally be extracted by the brain from the reference image. VIF is between the reference image and it's copy, is exactly unity. For all practical distortion types, VIF will lie in the interval [0,1].	1
FSIM	Feature Similarity Index measure is another objective image quality assessment measure that compares two images on the basis of low-level feature sets such as edges and zero-crossings. The working principle of FSIM is in line with our HVS which tends to understand images mainly on the basis of low level features. Also, any perceptible image degradations will lead to perceptible changes in these low-level features. The value of FSIM will lie in the interval [0,1].	1
SVDQA	It is an objective image quality assessment measure based on the use of singular values derived from singular value decomposition of the original image and the distorted image. It represents an error value obtained by computing the difference (or distance) between the singular values of the original image and the singular values of the distorted. The lower the value of SVDQA, higher is the similarity of the distorted image with the original image.	0
PLCC	Pearsons's Linear Correlation Coefficient Values range from –1 to +1. A value of –1 indicates perfect negative correlation, while a value of +1 indicates perfect positive correlation. A value of 0 indicates no correlation between two compared images.	1
KROCC	Kendall's Rank Order Correlation Coefficient Values range from –1 to +1.	1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	82	9	106	213	48	54	165	56	191	64	163	158	129	243	215	251
1	124	227	57	130	155	47	255	135	52	142	67	68	196	222	233	203
2	84	123	148	50	166	194	35	61	238	76	149	11	66	250	195	78
3	8	46	161	102	40	217	36	178	118	91	162	73	109	139	209	37
4	114	248	246	100	134	104	152	22	212	164	92	204	93	101	182	146
5	108	112	72	80	253	237	185	218	94	21	70	87	167	141	157	132
6	144	216	171	0	140	188	211	10	247	228	88	5	184	179	69	6
7	208	44	30	143	202	63	15	2	193	175	189	3	1	19	138	107
8	58	145	17	65	79	103	220	234	151	242	207	206	240	180	230	115
9	150	172	116	34	231	173	53	133	226	249	55	232	28	117	223	110
10	71	241	26	113	29	41	197	137	111	183	98	14	170	24	190	27
11	252	86	62	75	198	210	121	32	154	219	192	254	120	205	90	244
12	31	221	168	51	136	7	199	49	177	18	16	89	39	128	236	95
13	96	81	127	169	25	181	74	13	45	229	122	159	147	201	156	239
14	160	224	59	77	174	42	245	176	200	235	187	60	131	83	153	97
15	23	43	4	126	186	119	214	38	225	105	20	99	85	33	12	125

FIGURE 6. Figure showing 16 × 16 Inverse S-box decimal values used in our proposed approach.

the accuracy of the proposed approach after inspecting the resultant images and the meaningful shares using Human Visual System (HVS). The first column (i.e., (a)) in Figure 8 shows the original secret satellite images. The next three columns, i.e., columns (b)-(d) shows the three meaningless shares generated by the proposed approach for the secret satellite image shown in column (a). The three meaningful shares generated in our proposed work corresponding to the three meaningless shares (in columns (b)-(d)) are shown in columns (e)-(g). Here, down-sampled version of meaningful shares have been shown in the figure for sake of uniformity in displaying. The final recovered secret at the receiver side

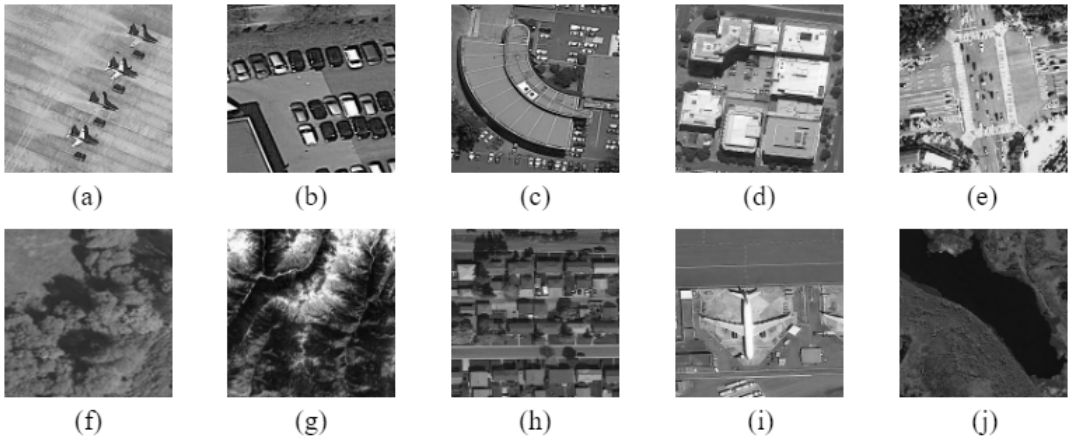

FIGURE 7. Figure showing ten randomly selected test satellite images for the demonstration of empirical results of proposed approach: (a)-(j) Ten test satellite images which are named as TestImage1 to TestImage10.

TABLE 2. Values of objective evaluation metrics between original and recovered secret (satellite) image on all 10 test images using our proposed approach.

	PSNR	MSSIM	FSIM	SVDQA	VIF	SROCC	KROCC
Test Image 1	Inf	1	1	0	1	1	1
Test Image 2	Inf	1	1	0	1	1	1
Test Image 3	Inf	1	1	0	1	1	1
Test Image 4	Inf	1	1	0	1	1	1
Test Image 5	Inf	1	1	0	1	1	1
Test Image 6	Inf	1	1	0	1	1	1
Test Image 7	Inf	1	1	0	1	1	1
Test Image 8	Inf	1	1	0	1	1	1
Test Image 9	Inf	1	1	0	1	1	1
Test Image 10	Inf	1	1	0	1	1	1

from the three meaningful shares (i.e., (e)-(g)) is shown in column (h). One can contemplate the high level of similarity garnered by the proposed approach between the set of images shown in column (a) and corresponding set of images shown in column (h).

Figure 9 shows the results in case of any intentional or unintentional attacks on the meaningful shares during transmission. The results are shown on three test satellite images displayed in column (a). Columns (b) and (c) displays three meaningless shares and three meaningful shares (down-sampled version for uniformity in displaying) generated from original secret (satellite) image (i.e., (a)), respectively. Column (d) represent three meaningful shares received by the receiver which got altered/tampered due to any intentional (or unintentional) attack during transmission. Images in column (e) are three meaningless shares extracted from tampered meaningful shares shown in column (d) and lastly, image (f) is the recovered secret from the tampered version of meaningless shares. Evidently from the Figure 9, it can be seen that the recovered secret (i.e., column (e)) is random in nature, substantiating the effectiveness of the proposed approach, such that, if any share got altered during transmission, then the contents of the secret cannot be revealed. The resultant image will be a garbage image as can be seen in column (e) of the figure.

Table 3 shows the imperceptibility achieved by our proposed approach between the original cover images and the three meaningful shares on all ten test images. The imperceptibility results are presented in terms of objective evaluation metrics described in Table 1. An average *PSNR* of $48dB$ between first cover image and first meaningful share, $47dB$ between second cover image and second meaningful share, and $46dB$ between third cover image and third meaningful share has been observed by the proposed approach over all the ten test images, which is quite satisfactory. The values of other metrics are also tending towards their ideal values. Higher values of *PSNR*, *MSSIM*, *FSIM*, *VIF*, *PLCC*, and *KROCC* obtained by the proposed approach indicates the efficacy of the proposed approach in concealing the meaningless shares in cover images such that our human visual system can not infer the possibility of something (secret) hidden in the meaningful shares. The meaningful shares were able to retain majority of the structural and low-level features of the cover images, resulting in profound similarity among them. This property is of paramount importance to avoid cryptanalysis. Table 4 illustrates the ability of the proposed approach to generate completely randomized (noise-like) meaningless shares with minutest of the correlation with the original secret image. In such comparisons, it is desirable that any VC approach

FIGURE 8. Figure showing the resultant images generated/extracted using our proposed approach: (a) Original Secret Image; (b)-(d) Three Meaningless Shares generated from original secret image (i.e., (a)); (e)-(g) Three Meaningful Shares generated from meaningless shares (i.e., (b)-(d)) (Down-sampled version for uniformity in displaying); (h) Recovered Secret by receiver from three meaningful shares (i.e., (e)-(g)).

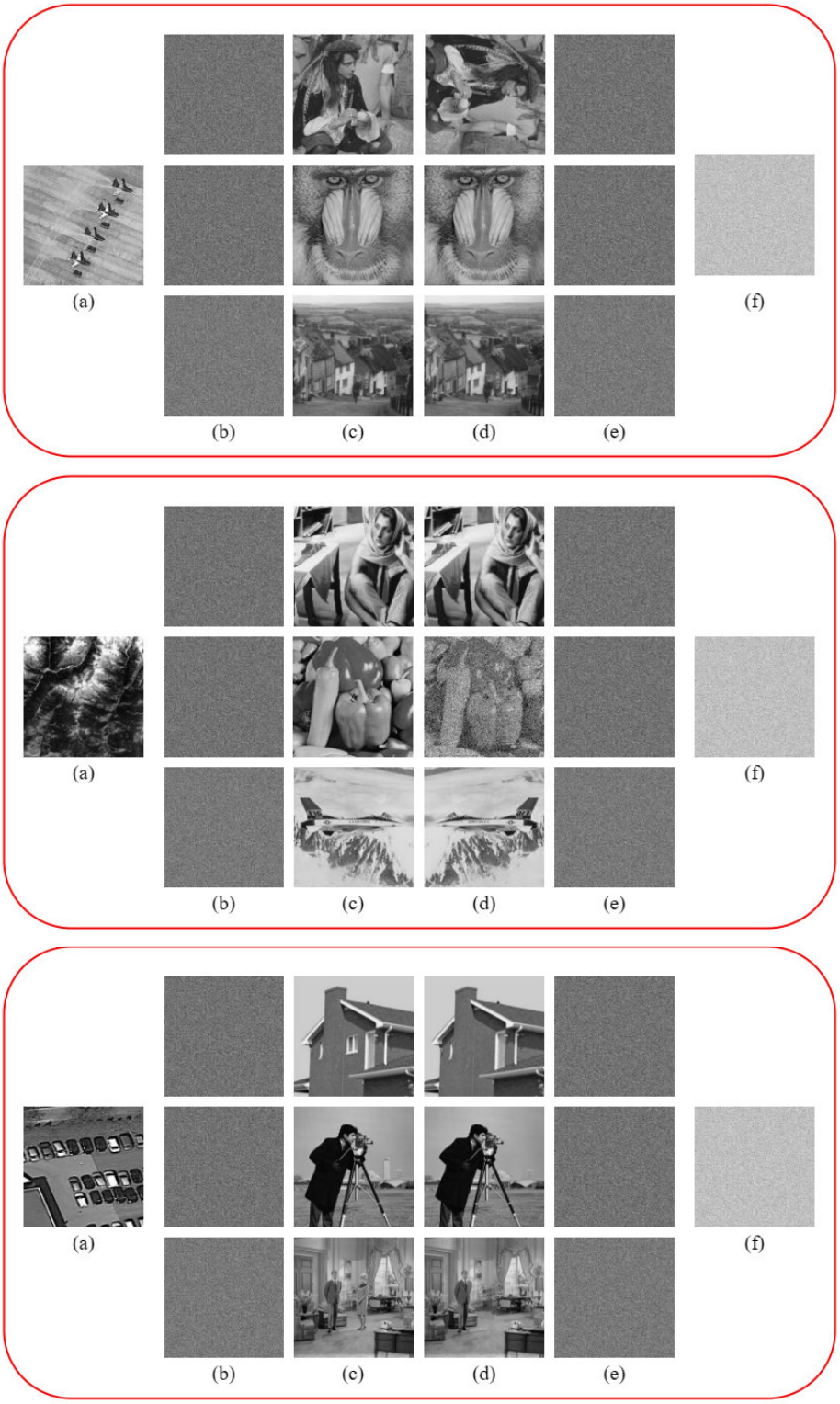

FIGURE 9. Figure showing the resultant images generated/extracted using our proposed approach: (a) Original Secret Image; (b)Three Meaningless Shares generated from original secret image (i.e., (a)); (c)Three Meaningful Shares generated from meaningless shares (i.e., (b)) (Downsampled version for uniformity in displaying); (d) Three Meaningful Shares received by the receiver which got altered/tampered due to intentional (or unintentional) attack (Downsampled version for uniformity in displaying);(e) Three Meaningless Shares generated from meaningful shares (i.e., (d)) at receiver side; (e) Recovered Secret which is a random image from three meaningful shares.

TABLE 3. Results showing the imperceptibility achieved using our proposed approach between original cover images and three meaningful shares on all 10 test images.

		PSNR	MSSIM	FSIM	SVDQA	VIF	PLCC	KROCC
Test Image 1	Share 1	48.80	0.9980	0.9994	3.9470	0.9861	0.9996	0.9960
	Share 2	47.26	0.9953	0.9985	3.6963	0.9915	0.9990	0.9947
	Share 3	46.49	0.9931	0.9990	4.1447	0.9899	0.9992	0.9958
Test Image 2	Share 1	48.78	0.9949	0.9993	3.5890	0.9928	0.9990	0.9957
	Share 2	47.34	0.9985	0.9997	4.0868	0.9922	0.9991	0.9973
	Share 3	46.84	0.9925	0.9998	3.5835	0.9940	0.9997	0.9982
Test Image 3	Share 1	48.57	0.9964	0.9990	3.7685	0.9939	0.9996	0.9986
	Share 2	47.83	0.9947	0.9983	4.0256	0.9883	0.9993	0.9982
	Share 3	46.63	0.9965	0.9983	4.1802	0.9920	0.9999	0.9987
Test Image 4	Share 1	48.40	0.9964	0.9985	3.4811	0.9870	0.9990	0.9925
	Share 2	47.79	0.9925	0.9996	4.3294	0.9853	0.9994	0.9949
	Share 3	46.58	0.9916	0.9985	4.1757	0.9924	0.9993	0.9944
Test Image 5	Share 1	48.40	0.9980	0.9995	3.8868	0.9900	0.9997	0.9987
	Share 2	47.84	0.9982	0.9985	3.8359	0.9898	0.9997	0.9986
	Share 3	46.34	0.9986	0.9998	3.8468	0.9940	0.9992	0.9972
Test Image 6	Share 1	48.55	0.9969	0.9987	3.7063	0.9911	0.9994	0.9982
	Share 2	47.73	0.9985	0.9984	3.9085	0.9912	0.9994	0.9984
	Share 3	46.26	0.9953	0.9985	3.9108	0.9936	0.9996	0.9986
Test Image 7	Share 1	48.25	0.9978	0.9992	4.2176	0.9931	0.9996	0.9968
	Share 2	47.30	0.9943	0.9989	4.1948	0.9908	0.9997	0.9971
	Share 3	46.35	0.9982	0.9987	4.0443	0.9868	0.9992	0.9961
Test Image 8	Share 1	48.79	0.9967	0.9996	3.7786	0.9874	0.9996	0.9976
	Share 2	47.50	0.9940	0.9991	4.2116	0.9939	0.9996	0.9971
	Share 3	46.82	0.9923	0.9990	3.9328	0.9853	0.9991	0.9967
Test Image 9	Share 1	48.64	0.9962	0.9997	3.7507	0.9899	0.9991	0.9959
	Share 2	47.57	0.9926	0.9985	4.3390	0.9867	0.9994	0.9959
	Share 3	46.85	0.9917	0.9994	4.2759	0.9948	0.9999	0.9968
Test Image 10	Share 1	48.44	0.9970	0.9994	3.9502	0.9921	0.9993	0.9939
	Share 2	47.64	0.9935	0.9987	4.0225	0.9900	0.9995	0.9955
	Share 3	46.59	0.9975	0.9991	3.9870	0.9897	0.9992	0.9943

TABLE 4. Objective Evaluation Measures between Original Secret Image and Meaningless Shares on all 10 Test Images.

		PSNR	MSSIM	FSIM	SVDQA	VIF	PLCC	KROCC
Test Image 1	Share 1	10.06	0.0191	0.5625	27.6301	0.00194	0.00781	0.00021
	Share 2	10.02	0.0182	0.5626	28.7267	0.00185	0.00743	0.00011
	Share 3	9.49	0.0154	0.5628	26.0404	0.00198	0.00622	0.00008
Test Image 2	Share 1	11.27	0.0283	0.5448	22.7931	0.00267	0.00472	0.00024
	Share 2	10.62	0.0270	0.5449	24.6283	0.00263	0.00479	0.00031
	Share 3	10.07	0.0201	0.5439	28.2176	0.00262	0.00457	0.00011
Test Image 3	Share 1	9.19	0.0285	0.5517	25.5866	0.00280	0.00145	0.00040
	Share 2	10.34	0.0257	0.5495	29.2921	0.00286	0.00290	0.00070
	Share 3	10.92	0.0278	0.5400	24.7003	0.00295	0.00243	0.00062
Test Image 4	Share 1	11.14	0.0223	0.5192	21.2252	0.00174	0.00146	0.00071
	Share 2	10.39	0.0209	0.5212	23.8553	0.00177	0.00150	0.00076
	Share 3	9.02	0.0166	0.5277	21.1828	0.00186	0.00145	0.00067
Test Image 5	Share 1	11.82	0.0282	0.5229	22.6192	0.00206	0.00498	0.00028
	Share 2	10.96	0.0255	0.5235	25.1099	0.00210	0.00469	0.00023
	Share 3	9.54	0.0191	0.5253	22.0413	0.00221	0.00397	0.00019
Test Image 6	Share 1	10.76	0.0246	0.5180	28.2511	0.00254	0.00760	0.00069
	Share 2	10.60	0.0223	0.5176	25.7519	0.00257	0.00729	0.00067
	Share 3	9.95	0.0176	0.5145	24.1530	0.00262	0.00619	0.00057
Test Image 7	Share 1	10.36	0.0319	0.5420	29.6466	0.00322	0.00499	0.00021
	Share 2	10.28	0.0289	0.5355	27.8354	0.00334	0.00611	0.00035
	Share 3	9.84	0.0223	0.5128	25.2247	0.00351	0.00653	0.00053
Test Image 8	Share 1	10.16	0.0204	0.5754	23.9559	0.00216	0.00243	0.00071
	Share 2	10.38	0.0189	0.5791	27.9920	0.00210	0.00148	0.00059
	Share 3	10.15	0.0148	0.5893	27.8129	0.00207	0.00123	0.00047
Test Image 9	Share 1	11.05	0.0536	0.5831	24.6062	0.00459	0.00598	0.00022
	Share 2	10.33	0.0474	0.5733	23.0350	0.00463	0.00600	0.00032
	Share 3	9.14	0.0342	0.5399	20.7511	0.00466	0.00557	0.00020
Test Image 10	Share 1	10.92	0.0355	0.5550	27.6637	0.00268	0.00625	0.00058
	Share 2	10.66	0.0318	0.5517	26.2014	0.00263	0.00645	0.00064
	Share 3	9.85	0.0233	0.5411	23.8400	0.00263	0.00648	0.00069

should exhibit lowest possible values of objective evaluation metrics *PSNR*, *MSSIM*, *FSIM*, *VIF*, *PLCC*, and *KROCC*, signifying it's ability to elude any attacker by preserving the contents of the original secret image without even indicating

TABLE 5. Comparative analysis of our proposed approach with state-of-the-art existing approaches on the basis of subjective and objective evaluation criteria.

	[28]	[26]	[31]	[25]	[27]	[24]	Proposed Approach	Ideal Values
Encryption	×	×	×	×	×	✓	✓	–
Meaningful Shares	×	✓	✓	×	✓	×	✓	–
Self-Authentication	×	×	×	✓	✓	×	✓	–
Share Generation Technique	Multi-prime modular arithmetic + Chinese Remainder Theorem	Simple modular arithmetic + Polynomial Based	XOR Based + LSB Stuffing	XOR Based	Hybrid Fractal Matrix	XOR Based + LSB Substitution	Newton Polynomial Based	–
Average PSNR (Meaningless)	10.79	10.48	11.00	10.38	10.28	10.76	10.31	0
Average MSSIM (Meaningless)	0.0344	0.0336	0.0349	0.0334	0.0331	0.0318	0.0257	0
Average FSIM (Meaningless)	0.6171	0.6008	0.6280	0.5954	0.5899	0.5627	0.5444	0
Average SVDQA (Meaningless)	28.7617	28.0113	29.2620	27.7612	27.5111	26.2605	25.3456	∞
Average VIF (Meaningless)	0.00353	0.00345	0.00359	0.00343	0.00340	0.00327	0.00267	0
Average PLCC (Meaningless)	0.00546	0.00532	0.00556	0.00528	0.00523	0.00499	0.00470	0
Average KROCC (Meaningless)	0.00051	0.00050	0.00052	0.00050	0.00049	0.00047	0.00044	0
Average PSNR (Meaningful)	45.19	43.76	46.14	43.29	42.81	40.43	47.57	∞
Average MSSIM (Meaningful)	0.9462	0.9163	0.9661	0.9064	0.8964	0.8466	0.9960	1
Average FSIM (Meaningful)	0.9491	0.9191	0.9690	0.9091	0.8991	0.8492	0.9990	1
Average SVDQA (Meaningful)	4.7622	4.6434	4.8414	4.6038	4.5642	4.3662	3.9602	0
Average VIF (Meaningful)	0.94098	0.91127	0.96079	0.90136	0.89146	0.84193	0.99052	1
Average PLCC (Meaningful)	0.94946	0.91948	0.96945	0.90948	0.89949	0.84952	0.99941	1
Average KROCC (Meaningful)	0.94676	0.91686	0.96669	0.90690	0.89693	0.84710	0.99661	1

IEEE Access

a slightest of the clue about it. The proposed approach has demonstrated remarkable performance in this aspect as can be seen from Table 4, with average correlation values (over all ten test images) of 0.00477, 0.00486, and 0.00446 using *PLCC* and 0.00043, 0.00047, and 0.00041 using *KROCC* correlation metrics for first, second and third meaningless share, respectively. These values signify zero or no correlation among the original secret image and the generated meaningless shares.

The proposed approach is also compared with several existing and recently published state-of-the-art VC approaches of similar nature on the basis of various subjective as well as objective evaluation parameters. This comparative analysis is shown in the Table 5. From the table it can be observed that the proposed approach is the only approach among the compared ones that offer triad of novel methods for Encryption, grayscale Meaningless Shares Generation, grayscale Meaningful Shares(concealing grayscale Meaningless Shares) Generation, and Self-Authentication. Rest of the compared methods lacks in offering one or other such features in their methodology. Primarily it has been observed that only handful of approaches offer the ability to conceal complete grayscale meaningless shares in grayscale natural images to generate grayscale meaningful images. Majority of the approaches either generate halftone meaningless shares or would conceal grayscale meaningless shares in color cover images. The proposed approach is unique in itself, providing all security requirements like confidentiality, authentication and integrity verification, through three-layer encryption, hash-based integrity verification, meaningful share generation, etc. Also, the tabulated results clearly elucidates that the proposed approach has substantially outperformed the other approaches on all the evaluation criteria.

V. CONCLUSION

This paper presents a state-of-the-art solution in the form of a self-authenticating multi-tone secret sharing visual cryptography scheme with meaningful shares, to tackle the challenging problem of secure transmission of satellite images over the network. The proposed approach addresses this problem by incorporating several novel characteristics such as, three layers of encryption, hash-based integrity verification, Newton polynomials-based grayscale meaningless shares construction, innovative *SPICE* (secure pixel integration for covert embedding) method and $EX - CEP$ (extraction of covertly embedded pixels) method for meaningful shares generation and meaningless shares extraction at the sender and receiver ends, respectively. In the proposed approach, the satellite image is securely transmitted in the form of three meaningful shares. The use of meaningful shares induces the sense of robustness against any attempts of cryptanalysis during transmission. In addition, the proposed approach satisfies the essential security requirements, such as confidentiality, authentication, and integrity verification. This is accomplished by employing a hash value of 128-bits and making use of three symmetric keys that are shared between the sender and the receiver. The effectiveness of the proposed method has been proven to be superior to that of existing state-of-the-art VC approaches through extensive experimentation as well as detailed comparative analysis employing a variety of evaluation metrics such as *PSNR*, *MSE*, *MSSIM*, *VIF*, *FSIM*, *SVDQA*, *PLCC*, and *KROCC*s. Based on the results of the experiments, it can be substantiated that the proposed method exhibits the best imperceptibility results between the cover images and the meaningful shares, across all the evaluation criteria. Also, in terms of randomness of the (generated) meaningless shares with respect to the original secret image, the proposed approach outperformed all the compared methods by exhibiting the lowest empirical values comparable to the ideal values expected from a VC approach across all the evaluation metrics. This, indicates the prowess of the proposed approach in ensuring the secure transmission of satellite image by preserving it's contents in noise-like meaningless shares, leaving no discernible trace.

In our future works, will would likely be focusing on enhancing encryption techniques and creating more meaningful shares. Advanced encryption methods are the need of the hour to provide stronger security against emerging threats, incorporating advancements in quantum-resistant algorithms or homomorphic encryption to preserve privacy in distributed systems. Additionally, we would work towards generating more meaningful shares that convey more information or possess additional properties, such as resilience to noise or distortion, enabling secure communication in diverse environments. Lastly, we would also focus on employing third party key management entity instead of sharing the keys directly between sender and receiver to further enhance the robustness to cryptanalysis.

REFERENCES

[1] E. Blanc and I. Noy, "Impacts of droughts and floods on agricultural productivity in new Zealand as measured from space," *Environ. Res., Climate*, vol. 2, no. 3, Sep. 2023, Art. no. 035001.

[2] M. Krichen, M. S. Abdalzaher, M. Elwekeil, and M. M. Fouda, "Managing natural disasters: An analysis of technological advancements, opportunities, and challenges," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 99–109, Sep. 2024.

[3] D. J. Rogers, S. E. Randolph, R. W. Snow, and S. I. Hay, "Satellite imagery in the study and forecast of malaria," *Nature*, vol. 415, no. 6872, pp. 710–715, Feb. 2002.

[4] M. Burke, A. Driscoll, D. B. Lobell, and S. Ermon, "Using satellite imagery to understand and promote sustainable development," *Science*, vol. 371, no. 6535, 2021, Art. no. eabe8628.

[5] *India Meteorological Department, Ministry Of Earth Sciences, Government Of India*. Accessed: Feb. 10, 2024. [Online]. Available: https://mausam.imd.gov.in/

[6] *Marcin Frackiewicz, TS2 Spcace*. Accessed: Feb. 10, 2024. [Online]. Available: https://ts2.pl/en/the-importance-of-satellite-imaging-for-national-security/

[7] S. Shivani, S. Agarwal, and J. S. Suri, *Handbook of Image-based Security Techniques*. Boca Raton, FL, USA: CRC Press, 2018.

[8] P. Punithavathi and S. Geetha, "Visual cryptography: A brief survey," *Inf. Secur. J., Global Perspective*, vol. 26, no. 6, pp. 305–317, Nov. 2017.

[9] D. R. Ibrahim, J. S. Teh, and R. Abdullah, "An overview of visual cryptography techniques," *Multimedia Tools Appl.*, vol. 80, nos. 21–23, pp. 31927–31952, Sep. 2021.

[10] G. Manikandan, R. Kumar, and N. Rajesh, "Image security using visual cryptography," in *Handbook of Research on Computer Vision and Image Processing in the Deep Learning Era*. Hershey, PA, USA: IGI Global, 2023, pp. 281–292.
[11] P.-L. Chiu and K.-H. Lee, "Efficient constructions for progressive visual cryptography with meaningful shares," *Signal Process.*, vol. 165, pp. 233–249, Dec. 2019.
[12] *Mersenne Twister Generator*. Accessed: Dec. 15, 2023. [Online]. Available: http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/emt.html
[13] *Philox Generator*. Accessed: Dec. 15, 2023. [Online]. Available: https://in.mathworks.com/help/MATLAB/ref/randstream.randstream.list.html
[14] *Rijndael S-Box*. Accessed: Dec. 25, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Rijndael_S-box
[15] M. Naor and A. Shamir, "Visual cryptography," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, Rome, Italy. Berlin, Germany: Springer, May 1994, pp. 1–12.
[16] G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, "Visual cryptography for general access structures," *Inf. Comput.*, vol. 129, no. 2, pp. 86–106, Sep. 1996.
[17] M. Nakajima and Y. Yamaguchi, "Extended visual cryptography for natural images," *J. WSCG*, vol. 10, nos. 1–2, pp. 303–310, 2002. [Online]. Available: https://otik.uk.zcu.cz/handle/11025/5993
[18] K. Jainthi, "A novel cryptographic technique that emphasis visual quality and efficieny by Floyd steinberg error diffusion method," *Int. J. Res. Eng. Technol.*, vol. 4, no. 2, pp. 428–439, Feb. 2015.
[19] C.-C. Chang, P.-Y. Lin, Z. H. Wang, and M. C. Li, "A sudoku-based secret image sharing scheme with reversibility (invited paper)," *J. Commun.*, vol. 5, no. 1, pp. 5–12, Jan. 2010.
[20] Y. Liu and C.-C. Chang, "A turtle shell-based visual secret sharing scheme with reversibility and authentication," *Multimedia Tools Appl.*, vol. 77, no. 19, pp. 25295–25310, Oct. 2018.
[21] H.-D. Yuan, "Secret sharing with multi-cover adaptive steganography," *Inf. Sci.*, vol. 254, pp. 197–212, Jan. 2014.
[22] T.-F. Cheng, C.-C. Chang, and L. Liu, "Secret sharing: Using meaningful image shadows based on gray code," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 9337–9362, Apr. 2017.
[23] J. He, W. Lan, and S. Tang, "A secure image sharing scheme with high quality stego-images based on steganography," *Multimedia Tools Appl.*, vol. 76, no. 6, pp. 7677–7698, Mar. 2017.
[24] Z. Zhou, C.-N. Yang, Y. Cao, and X. Sun, "Secret image sharing based on encrypted pixels," *IEEE Access*, vol. 6, pp. 15021–15025, 2018.
[25] O. S. Srujana, N. C. Mhala, and A. R. Pais, "Verifiable XOR-based visual secret sharing scheme for hyperspectral images," *J. Appl. Remote Sens.*, vol. 15, no. 1, Feb. 2021, Art. no. 016510.
[26] C.-N. Yang, P.-Y. Tsai, and Y. Liu, "A (k, n) secret document sharing with meaningful shares," *J. Inf. Secur. Appl.*, vol. 62, Nov. 2021, Art. no. 102973.
[27] K. Gao, J.-H. Horng, and C.-C. Chang, "An authenticatable (2, 3) secret sharing scheme using meaningful share images based on hybrid fractal matrix," *IEEE Access*, vol. 9, pp. 50112–50125, 2021.
[28] C.-N. Yang, C.-E. Zheng, M.-C. Lu, and X. Wu, "Secret image sharing by using multi-prime modular arithmetic," *Signal Process.*, vol. 205, Apr. 2023, Art. no. 108882.
[29] D. Zhang, M. Shafiq, L. Wang, G. Srivastava, and S. Yin, "Privacy-preserving remote sensing images recognition based on limited visual cryptography," *CAAI Trans. Intell. Technol.*, vol. 8, no. 4, pp. 1166–1177, Dec. 2023.
[30] B. Bernard Ehuil, C. Chen, S. Wang, H. Guo, and J. Liu, "A secure mutual authentication protocol based on visual cryptography technique for IoT-cloud," *Chin. J. Electron.*, vol. 33, no. 1, pp. 43–57, Jan. 2024.
[31] A. S. Rawat, M. Deshmukh, and M. Singh, "Natural share-based lightweight (n, n) single secret image sharing scheme using LSB stuffing for medical images," *J. Supercomputing*, vol. 79, pp. 19138–19167, Nov. 2023.
[32] M. Yan, Y. Hu, and H. Zhang, "Progressive meaningful visual cryptography for secure communication of grayscale medical images," *Multimedia Tools Appl.*, vol. 83, no. 11, pp. 33639–33652, Sep. 2023.
[33] X.-W. Wu and T.-H. Chen, "Security enhancement of an (n, n) threshold non-expansible XOR-based visual cryptography with unique meaningful shares," *Multimedia Tools Appl.*, vol. 83, no. 10, pp. 28913–28926, Sep. 2023.
[34] S. Saha, A. Kumar Chattopadhyay, A. Kumar Barman, A. Nag, and S. Nandi, "Secret image sharing schemes: A comprehensive survey," *IEEE Access*, vol. 11, pp. 98333–98361, 2023.

SUCHITA SHARMA received the B.Tech. degree in computer science and engineering from the Green Hills Engineering College, Solan, Himachal Pradesh Technical University (HPTU) Hamirpur, India, in 2015, and the M.E. degree in computer science and engineering from the Thapar Institute of Engineering & Technology, Patiala, India, in 2018, where she is currently pursuing the Ph.D. degree in computer science and engineering. Her research interests include image watermarking, medical image retrieval, and machine learning.

SHIVENDRA SHIVANI received the master's and Ph.D. degrees from the National Institute of Technology Allahabad, India, with visual cryptography and watermarking as an area of interest. He is currently an Assistant Professor with the Thapar Institute of Engineering & Technology, Patiala. His current research interests include gaming and animation, digital watermarking, pattern recognition, computer vision, algorithms, compression, biometrics, visual cryptography, and face recognition.

NITIN SAXENA received the M.Tech. degree in information security and the Ph.D. degree from MNNIT Allahabad, Prayagraj, Uttar Pradesh, India. He is currently an Assistant Professor with the Computer Science and Engineering Department, Thapar Institute of Engineering & Technology, Patiala, Punjab, India. His research interests include data clustering, nature inspired optimization techniques and their applications in digital image processing, and cloud computing.