Journal of Information Security and Applications 93 (2025) 104137

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Hiding information in encrypted images with (*t, n*) secret sharing for IoT and cloud services

Yijie Lin [a], Chia-Chen Lin [b,*], Ching-Chun Chang [c], Chin-Chen Chang [a]

[a] Department of Information Engineering and Computer Science, Feng Chia University, NO.100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, Republic of China
[b] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan, Republic of China
[c] Information and Communication Security Research Center, Feng Chia University, NO.100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, Republic of China

ARTICLE INFO

Keywords:
Cloud services
Encrypted image
Internet of things
Reversible data hiding
Secret sharing
Security

ABSTRACT

Privacy and security concerns have emerged with the rapid advancement of information technology and the exponential growth in data storage and cloud services. This paper addresses these issues in the context of the Internet of Things (IoT) and cloud services. Focusing on reversible data hiding in encrypted images (RDHEI), the study presents an innovative scheme based on (*t, n*) secret sharing using an expandable magic matrix-based data hiding, which offers flexible security levels. The scheme is designed to withstand hacker attacks by effectively dispersing risks through secret sharing, dividing the data into multiple shares. This ensures that leaking fewer than *t* shares does not compromise the entire data and provides a flexible parameter scheme. The use of the expandable magic matrix enhances both the embedding capacity and security, demonstrating the robustness of the proposed RDHEI scheme in protecting data in the digital age. Furthermore, our approach encrypts images at the IoT gateway rather than at the cloud server, enabling content owners to assert ownership claims—an ability not available in previous schemes. Experimental results confirm that our scheme maintains a constant concealment capacity of up to 4 bits per pixel (bpp), while safeguarding the confidentiality of the hidden data and preserving the randomness of the generated shares.

1. Introduction

The rapid advancement of network technology has led to an unprecedented increase in the amount of data stored in various media (especially images), which has greatly affected professional fields and social interactions. A natural consequence of this trend is the widespread adoption of cloud services over traditional on-premises storage methods. Therefore, ensuring the privacy and security of data stored and transmitted through cloud platforms has become critical. This necessity is particularly acute in the areas of the Internet of Things (IoT) and cloud services, where emerging risks and challenges related to information security are becoming increasingly apparent. As these technologies continue to advance, the task of effectively protecting data from evolving threats has become an imperative.

Reversible data hiding (RDH) technology, as discussed in the literature [1–6], involves embedding confidential messages into cover media while ensuring that both the original content can be fully recovered and the hidden messages can be extracted after retrieval. This technology has a wide range of applications across various fields, including copyright watermarking, medical confidentiality, secure military communications, IoT security, and cloud services. Traditionally, cryptography [7] has been a practical solution for scenarios requiring enhanced protection. However, with the emergence of new use cases over the past decade - such as the need to protect not only patient data but also X-ray images containing sensitive patient information to prevent leaks - the integration of data encryption and reversible data hiding has led to the development of reversible data hiding in encrypted images (RDHEI) technology [8–12]. This integrated approach has become an effective means of bolstering security in both Internet of Things and cloud environments [13,14].

As cloud storage technology continues to evolve, an increasing number of providers are entering the market to meet the growing data demands of users. However, this expansion also makes cloud services more attractive targets for hackers. Even when encrypted images are uploaded to cloud platforms, they remain susceptible to tampering, which can prevent recipients from restoring the original image. In 2018, Wu et al. [15] proposed a solution that integrates reversible data hiding in encrypted images (RDHEI) and secret sharing (SS) to mitigate such attacks. Secret sharing, introduced by Shamir in 1979 [16], relies on a (*t,*

* Corresponding author.
E-mail addresses: p1263670@o365.fcu.edu.tw (Y. Lin), Ally.cclin@ncut.edu.tw (C.-C. Lin), ccc@fcu.edu.tw (C.-C. Chang), ccc@o365.fcu.edu.tw (C. Chang).

https://doi.org/10.1016/j.jisa.2025.104137

Available online 23 June 2025

n) scheme based on the Lagrangian interpolation algorithm. In 2002, Thien and Lin applied this concept to imaging by introducing secret image sharing (SIS) [17]. SIS-based encrypted imaging provides effective risk mitigation and deters tampering by malicious actors to some extent. In recent years, similar concepts have been incorporated into RDHEI and SIS technologies [18–21], finding applications in IoT environments such as secure medical image transmission, intelligent surveillance, and industrial data protection. In 2024, Xiong et al. [18] proposed a method that utilizes the pixel values of a palette-based secret image and the bit values of cover image pixels as polynomial coefficients to embed both the secret image and authentication data. In the same year, Wang et al. [19] employed an XOR operation in conjunction with chain obfuscation techniques to develop a secure image sharing scheme based on the Chinese Remainder Theorem, thereby significantly enhancing the security of image protection. Also in 2024, Jiang et al. [20] introduced a robust secret image sharing scheme resilient to JPEG recompression, founded on a stable block condition, which achieves a comprehensive balance among security, robustness, imperceptibility, and fidelity. In 2025, Xiong et al. [21] advanced a dual-SIS mechanism that performs two sequential rounds of secret sharing, effectively reinforcing both the security and efficiency of the image sharing process.

Scholars have proposed enhancement techniques to counter potential hacker threats and improve data integrity in these applications [22–24]. This approach significantly strengthens privacy protection and overall security for images uploaded to cloud platforms. Currently, it is primarily used in IoT devices and medical settings. Fig. 1 illustrates the encryption process applied to medical images uploaded to cloud platforms, typically managed by the IoT Gateway, such as a hospital's information department. Afterward, the encrypted medical image is divided into n shares, which are distributed to n cloud servers, either managed by the hospital or an authorized cloud service provider (CSP). Each cloud server can embed additional data into the received encrypted share, including timestamps, copyright information, or authentication notes for content protection. Finally, authorized hospitals, doctors, or patients can retrieve the reconstructed medical images from t stego shares.

According to the concept depicted in Fig. 1, shared storage is distributed across multiple cloud servers, which can be located either within a single region or across various regions. Cloud service providers have the capability to embed various types of information, such as timestamps and copyrights, into the shares they receive. However, hospitals lack the capability to directly embed sensitive ownership-related information, including hospital names, departments, physicians, patients, and specific constraints. In the event of a data breach caused by an attack on a cloud server or malicious collusion among multiple CSPs, hospitals are left at a disadvantage and cannot assert ownership of the leaked medical data. This is because, as shown in Fig. 1, data hiding occurs within the cloud server rather than at the IoT gateway. In other words, hospitals do not have the opportunity to embed relevant ownership information before medical images are transformed into shares and uploaded to the cloud platform. Since migration details cannot be disclosed to CSPs, the data embedding process must occur prior to uploading the data to the cloud server. To address this limitation, this paper proposes a scheme that allows hospitals to safeguard the ownership and security of medical images by embedding ownership information before encryption and sharing. Unlike existing methods, our approach ensures that hospitals maintain control over their data integrity and provenance, even after the images are stored and distributed across cloud servers. This enhancement reduces the risks associated with unauthorized access and malicious collusion among CSPs, providing a more robust security framework for medical data protection. Fig. 2 outlines our improvements to the IoT gateway depicted in Fig. 1. Building on the IoT gateway in Fig. 1, we first encrypt the raw medical image. After generating the encrypted image, the hospital's ownership information and other relevant data are embedded while generating shares using our designed mechanism. These n shares are then distributed to n cloud servers, ensuring the hospital retains ownership and the security of the medical data. Ultimately, when a medical image is required for diagnostic or insurance purposes, an authorized hospital, doctor, or patient can reconstruct the medical image from the t received shares.

To achieve our objective, this paper proposes a novel RDHEI based on (t, n) secret sharing using the expandable magic matrix scheme. Building upon the exploiting modification direction (EMD) algorithm introduced by Zhang et al. [25] in 2006, the EMD-2 algorithm proposed by Kim et al. [26] in 2010 derives the magic matrix. These algorithmic improvements have been made over the past decade or so. During this period, numerous matrix-based RDH methods [27–30] have been developed, and the utilization of matrices has effectively enhanced embedding capacity. The scheme we propose employs the Lagrangian interpolation algorithm to encrypt the original medical image and subsequently generates n shares based on (t, n) secret sharing for embedding the secret message. The recipient end only needs to possess t shares to reconstruct the encrypted image. Depending on the type of key in possession, it is possible to either reconstruct the original image or extract the secret message. The major contributions of this article mainly include the following aspects:

1. We define an architecture to empower medical images' owners, such as hospitals, by embedding the ownership or other additional information into shares before uploading shares to cloud servers. As

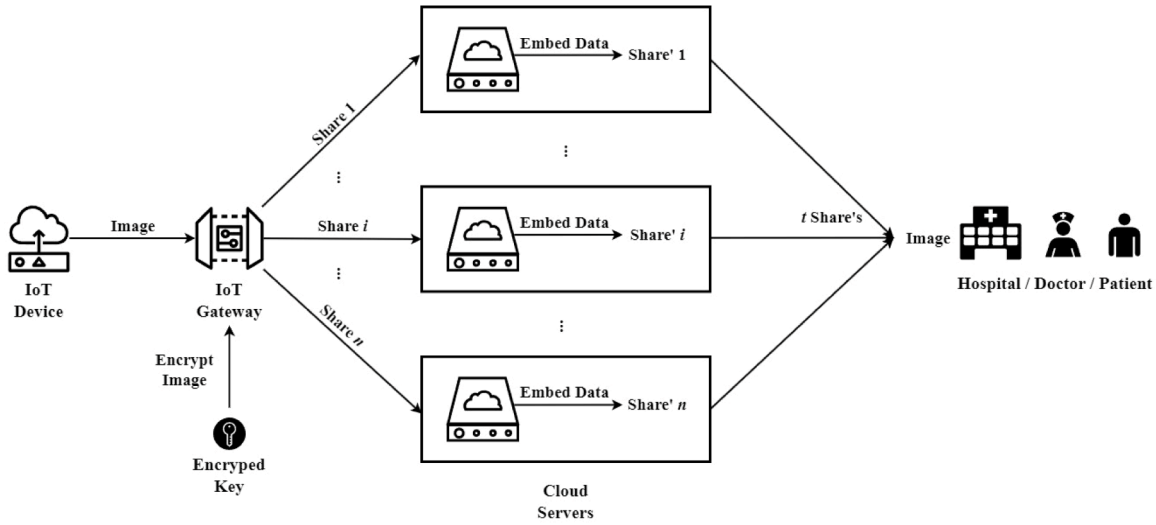

Fig. 1. Cloud secure storage and management of medical images for telemedicine consultation.

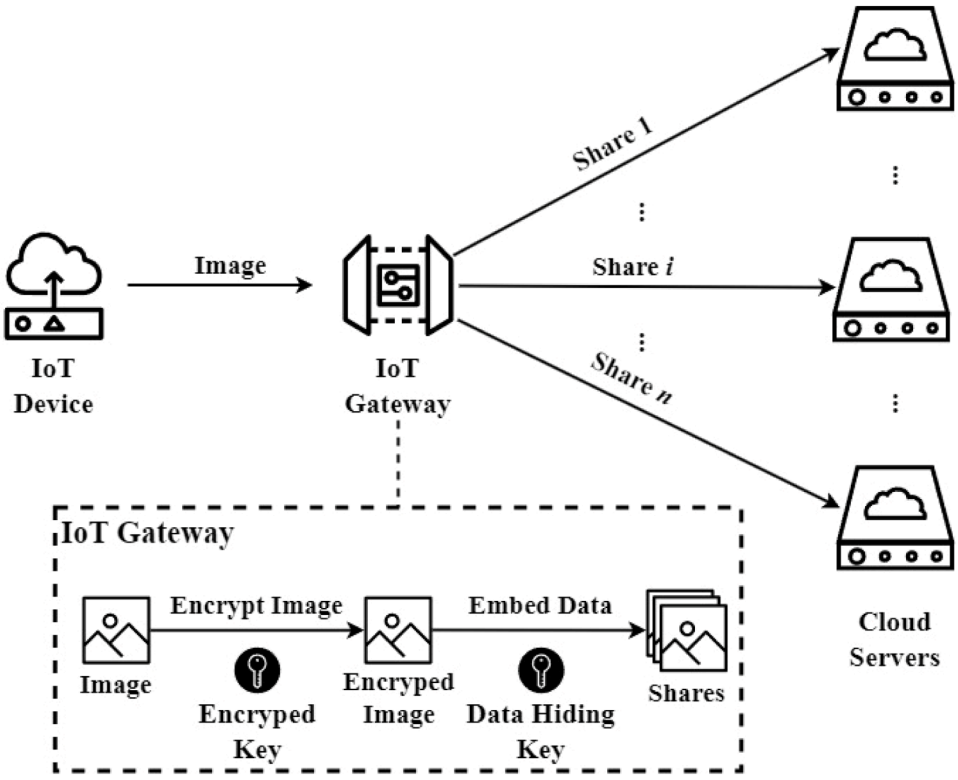

Fig. 2. Our proposed scheme improves the IoT gateway in Fig. 1.

long as t shares have been collected, the original image can be recovered completely and the owner is able to claim the ownership.

2. We utilize the expandable magic matrix. Based on the original magic matrix, the side length l of the search box can be adjusted according to the actual application scenario. The larger l is, the larger the embedded capacity is and the stronger the relative security. When $l \in [256, 257]$, the embedded capacity reaches 2097,152 bits. This demonstrates the excellence of our proposed scheme.

The subsequent sections of this paper are structured as follows: Section II reviews the related work, covering secret sharing and the magic matrix. Section III introduces our proposed RDHEI scheme, which is based on secret sharing utilizing the magic matrix. Section IV discusses the expandability of the magic matrix. Section V is devoted to presenting the experimental findings. Lastly, Section VI offers the conclusions of this paper.

2. Related work

In this section, we introduce two fundamental techniques in the subsequent subsections: Shamir's secret sharing [16] and Kim et al.'s magic matrix [26].

2.1. Secret sharing

In 1979, Shamir [16] proposed a (t, n) secret sharing scheme based on the Lagrangian interpolation algorithm, where a secret is divided into n shares and can only be reconstructed if at least t shares are available. This method is easy to implement, ensures strong security, and is widely applied in secure multi-party computation.

As depicted in Eq. (1), the secret message s is incorporated into an equation along with $(t-1)$ randomly chosen coefficients $a_1, a_2, \ldots, a_{t-1}$, where p is a prime number. This equation forms a curve, and its points are distributed as shares among n participants. Each participant receives a unique point $(x, f(x))$, ensuring that at least t must collaborate to recover the secret. Eq. (2) is used to derive the polynomial and compute $f(0) = s$, thereby revealing the secret message. The Lagrange interpolation polynomial, $L(x)$, is employed to reconstruct the original polynomial. Here, $x_j$represents the x-coordinate of the j-th share, r denotes the number of shares used in reconstruction, and t specifies the shares distinct from j.

$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \ (mod\ p), \tag{1}$$

$$L(x) = \sum_{j=1}^{r} \left(f(x_j) \times \prod_{\substack{0<t\le r \\ t\neq j}} \frac{x - x_t}{x_j - x_t} \right) \ (mod\ p). \tag{2}$$

Shamir's mechanism remains highly secure, even in the presence of attacks or information leaks, as long as the threshold t is not surpassed. It also offers flexibility through updates, allowing for the addition of new points to the original curve as fresh shares or the creation of a new curve from existing shares. This adaptability enables modifications to the threshold t and the invalidation of specific original shares, ensuring the continued robustness of the secret sharing scheme.

2.2. Magic matrix

Kim et al. [26] proposed the EMD-2 scheme based on Zhang and Wang's EMD [25] in 2010, which modifies at most two pixels of the LSB values to carry secrets. The data hiding and extraction operation are roughly described in Eqs. (3)-(7). As shown in Eq. (3), the basis vector $B_n([b_1, b_2, \ldots, b_n])$ is initially generated, and then Eq. (4) calculates the value of w in the $(2w+1)$-ary system, where n represents the number of cover pixels. Eq. (5) calculates the extraction function f, which can be represented as the inner product between an image pixel value vector $G_n([g_1, g_2, \ldots, g_n])$ and a basis vector B_n. The secret bits s is converted into secret digits in the $(2w+1)$-ary system by Eq. (6), where d is the number of the secret to be embedded. Finally, the cover pixel vector is modified by Eq. (7), where C_s is the coefficient vector corresponding to the number s associated with the basis vector B_n.

$$[b_1,\ b_2,\ \ldots,\ b_n] = \begin{cases} [1,\ 3], & n = 2 \\ [1,\ 2,\ 6,\ 11,\ 16,\ 21,\ \ldots,\ 6+5(n-3)], & n > 2 \end{cases}. \quad (3)$$

$$w = \begin{cases} 4, & n = 2 \\ 8+5(n-3), & n > 2 \end{cases}. \quad (4)$$

$$f(g_1, g_2,\ \ldots, g_n) = \left[\sum_{i=1}^{n} (g_i \times b_i)\right] \ (mod\ (2w+1)) \ . \quad (5)$$

$$s = \begin{cases} d-f, & if\ d \geq f \\ (2w+1)-|d-f|, & if\ d\langle f\ and\ n\rangle 2 \\ 4-|d-f|, & if\ d < f\ and\ n = 2 \end{cases}. \quad (6)$$

$$G_n^{'} = G_n + C_s \ . \quad (7)$$

Overall, when utilizing the magic matrix defined by EMD-2, there is a greater embedding capacity compared to EMD, while the level of image distortion remains nearly unchanged.

3. Proposed scheme

To facilitate the owner's claim of ownership of the original image while ensuring the security of uploaded images, we present a (t, n) secret sharing scheme based on the expandable magic matrix for reversible data hiding in encrypted images in this section. Here, (t, n) denotes that the secret message is embedded within the encrypted image, generating n shares. A minimum of t available shares are required to reconstruct the encrypted image, enabling the extraction of the secret message or the recovery of the original image, contingent upon the receiver's possession of the suitable key. Below, we will provide a detailed explanation of our scheme.

Fig. 3 depicts the flowchart of our scheme. RDHEI typically has three roles: content owner, data hider, and receiver. In our scheme, the content owner first uses the encryption key K_e to encrypt the original image, resulting in an encrypted image. Upon receiving the encrypted image, the data hider employs the embedding key K_d to embed the secret data into the encrypted image, while generating n shares. When the receiver possesses at least t shares, or when t distinct recipients collaborate with each other, the final encrypted image can be reconstructed. With the embedding key K_d, one representative recipient can extract the secret data. With the encryption key K_e, the original image can be recovered. If both the embedding key K_d and the encryption key K_e are available, both the secret data and the original image can be extracted and recovered.

3.1. Image encryption phase

We utilize Lagrangian polynomials for image encryption. Initially, each pair of adjacent pixels in the original image I is grouped into a 1×2 pixel pair. Let $I(a, b)$ represent a pixel pair, where a and b are the two pixel values, as defined in Eq. (8). The encryption key K_e is a randomly chosen number. Since the maximum pixel value in a grayscale image is 255, we select the smallest prime number greater than this range, which is 257. We compute $f(1)$ and $f(2)$ based on the given a and b, using these results to generate the transformed pixel pair. Once all pixel pairs in the

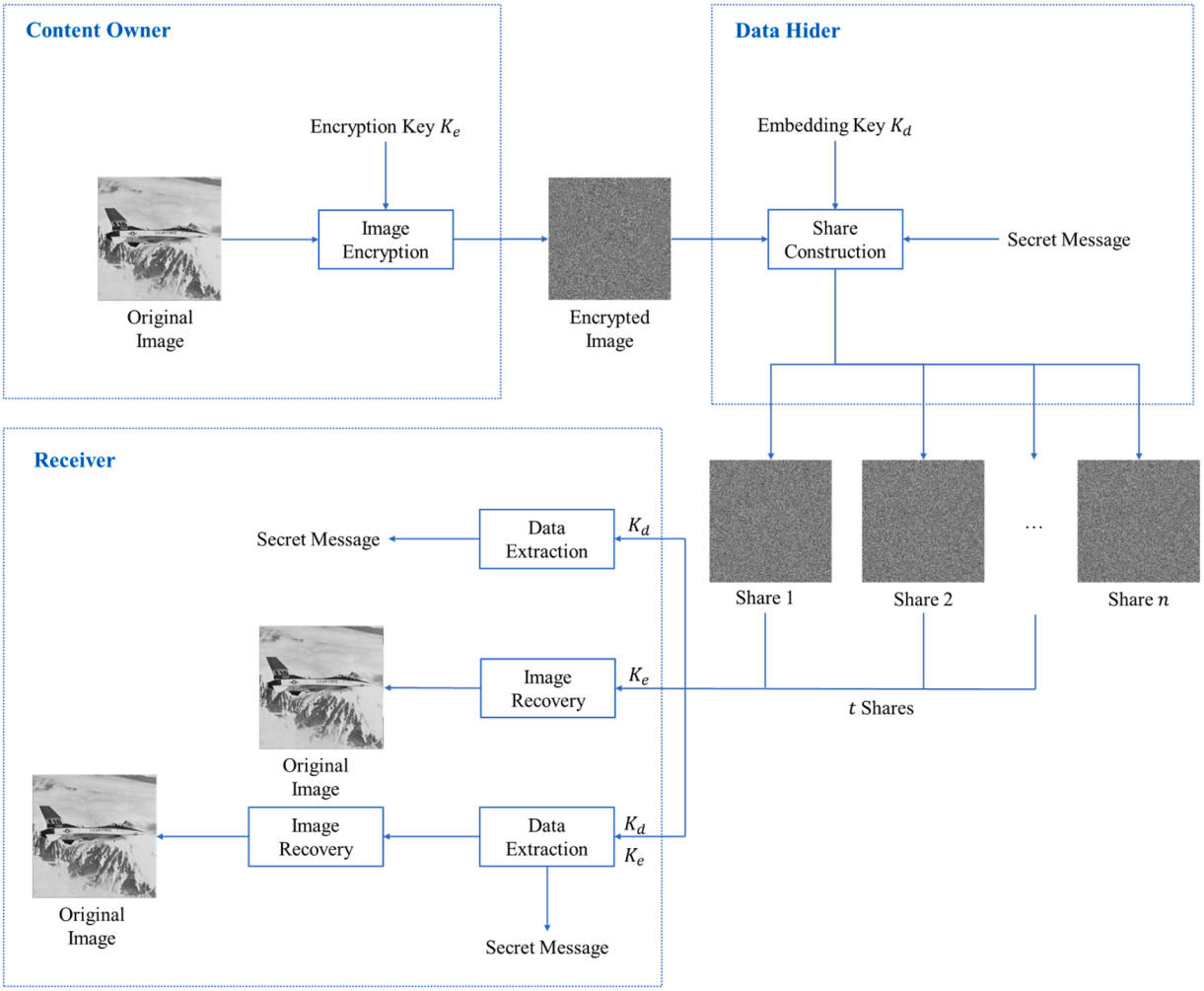

Fig. 3. Flowchart of our scheme.

original image have been processed, an intermediate encrypted representation is obtained. This representation acts as a preprocessing step for share construction rather than a final encrypted image. Consequently, values exceeding 8 bits, such as 256, are permissible at the image encryption stage. However, at the share construction stage, for the share, if $f(x)$ equals 256, we skip that value and proceed with the next x to ensure all shared pixels remain within the 0–255 range.

$$f(x) = ax^2 + bx + K_e \ (mod\ 257). \tag{8}$$

Fig. 4 illustrates an example, assuming that the pixel value of the original pixel pair (a, b) is (86, 255), and the encryption key is $K_e = 1721$. Equation $f(x) = 86x^2 + 255 + 1721(mod257)$can be constructed by Eq. (8). Subsequently, by calculating $f(1) = 6$ and $f(2) = 5$, we utilize them as the pixel values of the encrypted pixel pair to derive the encrypted pixel pair (6, 5).

3.2. Share construction phase

The secret sharing mechanism we propose relies on Kim et al.'s magic matrix [26]. Algorithm 1 delineates the construction phase of the expandable magic matrix. This algorithm is merely one of the methods for generating the matrix. In theory, it is imperative to ensure that all values in any $l \times l$ square of the matrix are unique, which satisfies the criteria of our matrix. Fig. 5 displays the original magic matrix introduced by Kim et al. [26], which is created based on Algorithm 1 when $l = 3$.

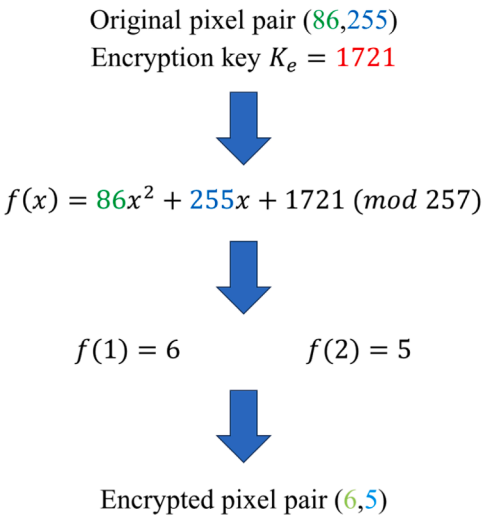

Fig. 4. Example of construct encrypted pixel pair with a given pixel pair of a cover image.

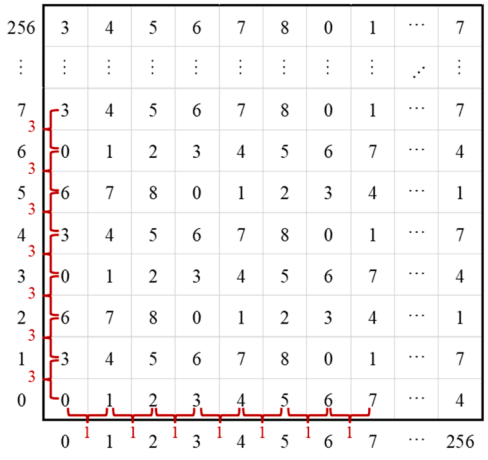

Fig. 5. Original magic matrix MM when $l = 3$.

Algorithm 1 Expandable Magic Matrix Construction.
Input: l: The side length of the search square
Output: MM: Expandable magic matrix
Step 1: Initialize MM as a 257 × 257 empty matrix.
Step 2: Set $MM(0, 0)$ to 0.
Step 3: Generate values for row 0 based on $MM(0, 0)$.
for j **from** 1 to 256 **do:**
$MM(0, j) = (MM(0, j - 1) + 1) mod l^2$
end for
Step 4: Generate values for each row based on the previous row.
for i **from** 1 to 256 **do:**
for j **from** 1 to 256 **do:**
$MM(i, j) = (MM(i - 1, j) + l) mod l^2$
end for
end for
Step 5: Output the expandable magic matrix MM.

Notably, the MM is sized as 257 × 257; this is because it is generally applicable to 8-bit depth images, and 257 is the smallest prime number greater than the pixel range of 0–255. For images with different bit depths, the matrix size can be modified accordingly to accommodate the new range.

Upon completing the image encryption, the secret information is embedded to generate shares. Algorithm 2 furnishes a comprehensive description of the share construction phase.

Algorithm 2 Share Construction.
Input: E: Encrypted image sized $w \times h$, M: Secret message, n: Number of share images, MM: Expandable magic matrix
Output: S_i: Share images $(1 \le i \le n)$
Step 1: Divide the secret message M into $\frac{w \times h}{2}$ segments of length $\lfloor log_2 l^2 \rfloor$ and convert them to decimal $m_k\left(1 \le k \le \frac{w \times h}{2}\right)$.
Step 2: Construct each pixel pair of share images.
for k **from** 1 to $\frac{w \times h}{2}$ **do:**
Step 2.1:
Set $MM(a_k,b_k)$ as the search center and perform a search in the area of $l \times l$.
Initialize$path_k = 0$.
while $MM(a_k',b_k') \neq m_k$ **do:**
Perform a clockwise spiral search.
$path_k = path_k + 1$
Update $MM(a_k',b_k')$ according to $path_k$.
- If the search square exceeds the MM boundary, shift it into the MM according to Fig. 8.
end while
Step 2.2: Construct an equation according to Eq. (9) for generating stego pixel pair of share images.
Step 2.3: Calculate values of $f_k(1), f_k(2), \ldots, f_k(2n)$.
Step 2.4: Construct each pixel pair of share images.
for i **from** 1 to n **do:**
$S_i(a_k,b_k) = (f_k(2i - 1), f_k(2i))$
end for
end for
Step 3: Output n share images $S_i(1 \le i \le n)$.

Eq. (9) is a crucial equation for constructing share images. In this equation, a and b represent the two pixels of the encrypted image pixel pair, $path$ represents the location of the secret message, and K_d is the embedding key, a random number. Here, t signifies the number of share images required to reconstruct the encrypted image. The polynomial consists of a total of $2t$ terms, among which the coefficients of the three smallest terms are $path \times K_d$, b and a, respectively. The coefficients of the remaining terms are represented by a random number r or multiple different random numbers r_1, r_2, and so forth.

$$f(x) = r_1 x^{2t-1} + \ldots + ax^2 + bx + (path \times K_d) \ (mod\ 257). \tag{9}$$

Fig. 6 gives a specific example when $l = 3$. Assuming that the pixel pair in the encrypted image is (6,5), and the secret information is $(101)_2$, which corresponds to the decimal representation of $(5)_{10}$. In the magic matrix, we construct a search square of size $l \times l$ with $MM(6, 5)$ as the center and conduct a clockwise spiral search. When $path = 6$, we search for the value 5, which corresponds to binary bitstream $(101)_2$.

In this example, we assume $(t, n) = (2, 2)$, on this basis we use Algorithm 2 to construct the share. In Fig. 7, we provide the specific calculation process. Continuing with the example provided in Fig. 5, we proceed to utilize the encrypted pixel pair (6,5) generated previously, with the embedding key $K_d = 3529$ and $path = 6$ as the coefficients of the minimum three terms. For the remaining terms, a random number r is employed as the coefficient; here, $r = 28703$. Consequently, we obtain the equation $f(x) = 28703x^3 + 6x^2 + 5x + (6 \times 3529)(mod257)$. Subsequently, we compute the values from $f(1)$ to $f(2n)$. Given that $n = 2$, we calculate the values of $f(1)$ to $f(4)$, resulting in 103, 49, 223, and 139 respectively. Finally, we ascertain that the pixel pair of Share 1 is (103,49), and the pixel pair of Share 2 is (223,139). After processing all pixel pairs in the image, $n = 2$ shared images are successfully generated. Unlike the encrypted image construction, the value 256 is not allowed in the share construction process. Therefore, if$f(x) = 256$ for any x within the current range (e.g., $x = 1, 2, 3, 4$), both x and $f(x)$ are discarded. The function is then evaluated for the next available$x$ values (e.g., $x = 5, 6, \ldots$) until all values fall within the valid range of 0–255.

The example provided above comprehensively illustrates the shared construction phase for $n = 2$. The given example is entirely within the matrix. Fig. 8 demonstrates an example where the search path commences from $MM(0, 0)$. In Fig. 8(a), a search square of size $l \times l$ is formulated with $MM(0, 0)$ as the center, where $l = 3$. It is evident that the search block extends beyond the bounds of the magic matrix and consequently necessitates relocation into the magic matrix. In Fig. 8(b), we address the beyond bounds issue as outlined in Algorithm 2, Step 2.1. Adjustments are made to the search blocks to ensure they fit within the magic matrix. In this instance, we continue to utilize $MM(0, 0)$, following a clockwise spiral pattern. If the subsequent step surpasses the magic matrix, as indicated by the dotted line in the figure, the search can proceed according to the clockwise spiral pattern until the search scope encompasses the matrix content once more. In this example, the search path proceeds as follows:$MM(0, 0) \rightarrow MM(1, 0) \rightarrow MM(0, 1) \rightarrow MM(1, 1) \rightarrow MM(2, 1) \rightarrow MM(2, 0) \rightarrow MM(0, 2) \rightarrow MM(1, 2) \rightarrow MM(2, 2)$.

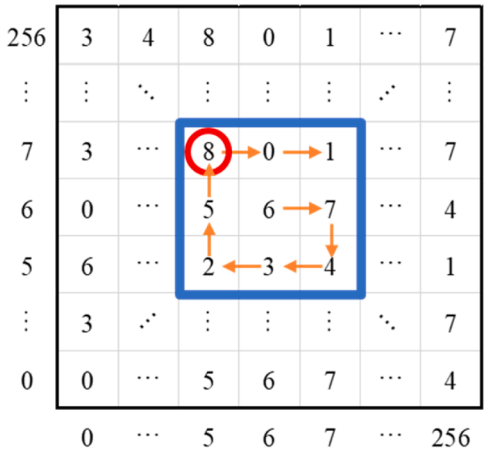

Fig. 6. Example of construct shares pixel pairs.

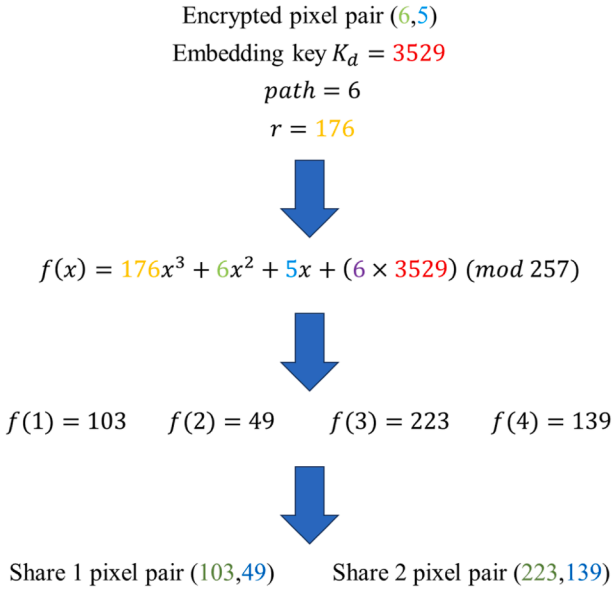

Fig. 7. Example of construct shares pixel pairs.

3.3. Data extraction and image recovery phase

For the receivers, they must initially possess t share images. Subsequently, if they possess the embedding key K_d, recipients have the capability to extract the hidden secret message. Conversely, if they possess the encryption key K_e, recipients can decrypt the encrypted image to retrieve the original image. If both the embedding key K_d and the encryption key K_e are available, recipients can extract the secret message and recover the original image simultaneously. Algorithm 3 furnishes a comprehensive description of this process.

Algorithm 3 Data Extraction and Image Recovery.
Input: S_i: Share images ($1 \le i \le t$)
Output: M: Secret message, I: Original image
Step 1: Divide all the pixels of t share images into pixel pairs of size 1×2.
Step 2: Reconstruct each pixel pair of the encrypted image.
for k **from** 1 to $\frac{w \times h}{2}$ **do:**
Step 2.1: Obtain function values.
for i **from** 1 to t **do:**
$f_k(2i - 1) = S_i(a_k)$
$f_k(2i) = S_i(b_k)$
end for
Step 2.2: Utilize the Lagrangian interpolation algorithm on all polynomial solutions to determine all coefficients of the polynomial.
end for
Step 3: Having different keys has different results.
if (the receivers have both embedding key K_d and encryption key K_e):
Perform Steps 3.1 and 3.2 simultaneously.
else if (the receivers have embedding key K_d):
Perform Steps 3.1.
else if (the receivers have encryption key K_e):
Perform Steps 3.2.
end if
Step 3.1: Extract secret message segment.
Initialize$M$ as "".
for k **from** 1 to $\frac{w \times h}{2}$ **do:**
Step 3.1.1: Determine the value of $path_k$ using the calculated coefficient $path_k \times K_d$ from Step 2.2.
Step 3.1.2:
Set $E(a_k,b_k)$ as the search center and perform a search in the area of $l \times l$.
Perform a clockwise spiral search according to $path_k$ to obtain $MM(a_k',b_k')$.
- If the search square exceeds the matrix boundary, shift it into the matrix according to Fig. 8.
$m_k = MM(a_k',b_k')$
Step 3.1.3: Convert m_k into binary form to obtain the secret message segment, and then splice the segments into a complete M.
$binary\_m_k = dec\_to\_bin(m_k)$
$M = M \parallel binary\_m_k$
end for
Step 3.2: Recover original image.
for k **from** 1 to $\frac{w \times h}{2}$ **do:**
In Step 2.2, obtain $E(a_k,b_k)$. Use the Lagrangian interpolation algorithm to solve for the coefficients of the encryption equation, thus obtaining the original pixel pair $I(a_k,b_k)$.
end for
Step 4:
if (the receivers have both embedding key K_d and encryption key K_e):
Output secret message M and original image I.
else if (the receivers have embedding key K_d):
Output secret message M.
else if (the receivers have encryption key K_e):
Output original image I.
end if

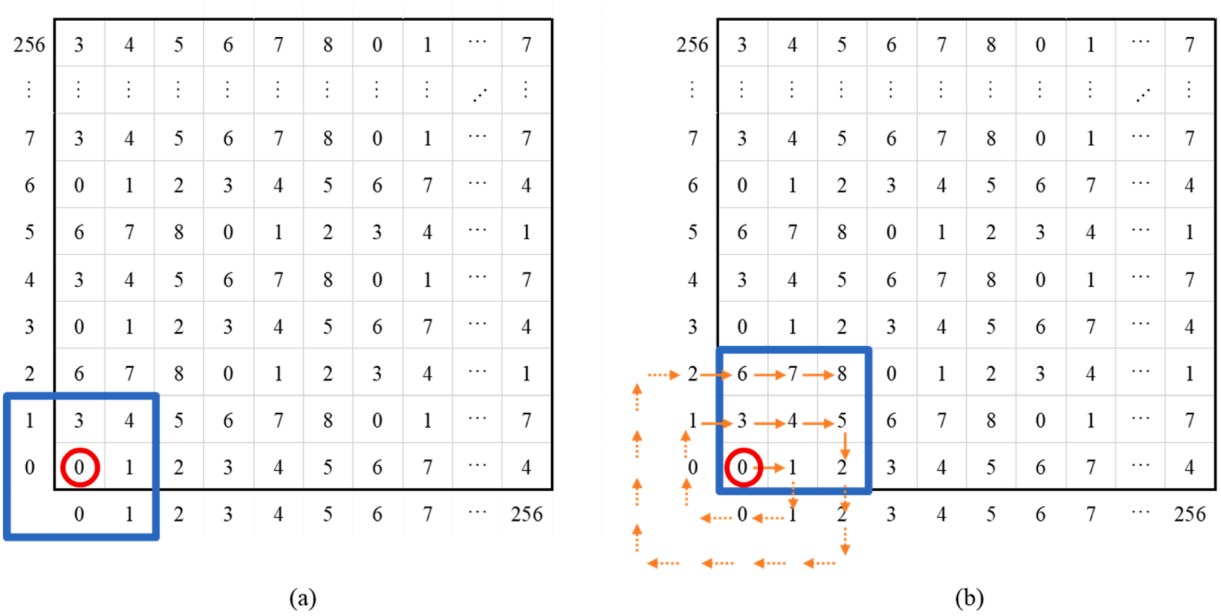

Fig. 8. Example of beyond bounds resolution: (a). Original search square (b) Modified search square.

We provide a detailed example of the data extraction and image recovery phase in Fig. 9. Similar to the share construction phase example, we proceed with $l = 3$ and $(t, n) = (2, 2)$. As a result, we begin with two share images, and consider a specific pixel pair as an example. In this case, pixel pair of Share 1 is (103, 49), and pixel pair of Share 2 is (223, 139). It's clear that $f(1)$ to $f(4)$ correspond to 103, 49, 223 and 139, respectively. Using the Lagrangian interpolation algorithm, we can determine each coefficient of the polynomial. Here, coefficients 6 and 5 represent the pixel values of the encrypted pixel pairs, while 100 represents $path \times K_d \equiv 100(mod257)$. If we know that the embedding key K_d is 3529, we can calculate that $path = 6$. Subsequently, as shown in Fig. 6, we initiate a clockwise spiral search for a square of size $l \times l$ centered on the location $MM(6, 5)$, which is based on the encrypted pixel pair $E(6, 5)$. During the search, we find the value 5 corresponding to $path = 6$. Converting this value to binary yields the secret message segment $m = (101)_2$. If we have the encryption key K_e, we can utilize the Lagrangian interpolation algorithm to determine the coefficients a and b in Eq. (8), which correspond to the values of the original pixel pair $I(248, 8)$. By recovering all original pixel pairs, we can reconstruct the original image. If we know both embedding key K_d and encryption key K_e, we can both extract the secret message and recover the original image at the same time.

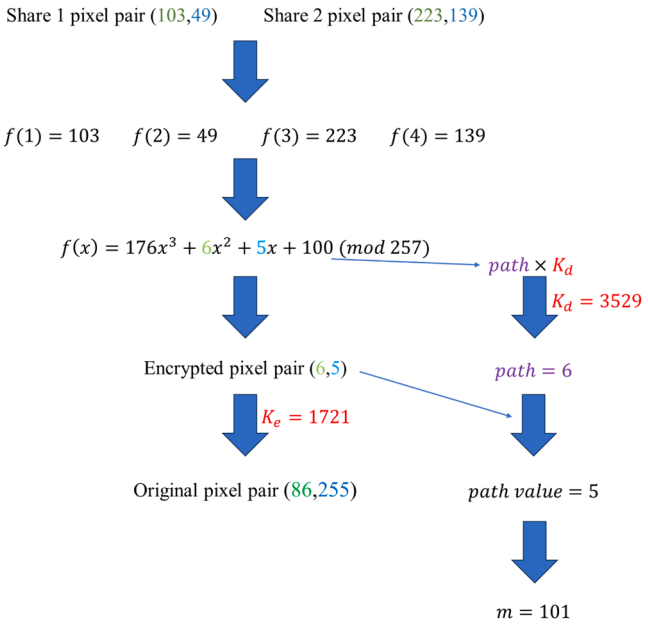

Fig. 9. Example of data extraction and image recovery.

4. Discussion on the expandability of magic matrix

This section explores the extension of the magic matrix concept for $l > 3$, building on the method introduced by Kim et al. in Section III. The primary motivation behind the extendable magic matrix is to enhance both embedding capacity and security. By increasing the size of the matrix, we allow more distinct values, thereby improving the overall security of the system. Fig. 10 provides an example with $l = 5$, where the elements range from 0 to 24, and each value is distinct within any 5×5 square.

Compared to the original magic matrix, the extended version incorporates more distinct values. For example, when $l = 5$, the elements range from 0 to 24, providing 16 more distinct values than when $l = 3$. This increase in distinct values significantly enhances security. With more elements, the probability of guessing the correct values decreases, making brute-force attacks increasingly impractical. In addition to boosting security, our scheme also increases the embedding capacity. The embedding capacity for each pixel pair is given by $\lfloor log_2 l^2 \rfloor$, meaning larger values of l result in higher embedding capacity. For clarity, while our examples focus primarily on cases where l is odd, it is important to note that l can also be even. Furthermore, the starting point of our search path is not restricted to the center of the search box, and during implementation, defining how to conduct searches within the domain is the main requirement.

As the side length l increases, the search range expands, improving security by providing more distinct values and configurations. This makes it harder for attackers to guess the correct configuration or use brute-force techniques. Additionally, as l grows, embedding capacity increases since each pixel pair can carry more information. While the computational complexity rises with a large l, it remains within the $O(n)$ range, meaning the performance impact is minimal. This demonstrates the scalability of our approach, allowing the side length l to be adjusted

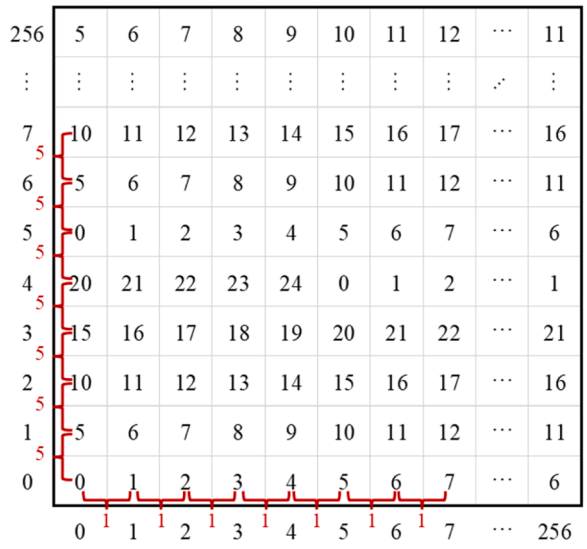

Fig. 10. Extended magic matrix when $l = 5$.

according to specific security and embedding requirements, thereby balancing embedding capacity, security, and computational efficiency.

5. Experimental results

This section focuses on evaluating the performance of our proposed expandable magic matrix-based (t, n) secret sharing for RDHEI, showcasing its superiority across multiple dimensions. Fig. 11 displays the six test images used in our experiment: "Airplane," "Baboon," "Boat," "Lake," "Office," and "Peppers." The experiment is conducted within an environment characterized by an AMD Ryzen 7 5800H CPU with Radeon Graphics operating at 3.20 GHz, supported by 16 GB of RAM running at 3200 MHz. The operating system utilized is Windows 10, with MATLAB R2023b serving as the primary software tool.

Our experiment is conducted in the case of $(t, n) = (2, 2)$. Fig. 12 shows the images and their histograms of our proposed scheme at different phases, where (a) represents the original image, (b) represents the encrypted image, (c) and (d) represent Share 1 and Share 2, respectively. Additionally, (e) represents the reconstructed encrypted image, (f) represents the reconstructed original image. We observe that the histograms of encrypted image and shares exhibit a very smooth distribution, indicating that our scheme is highly random and effectively reflects its security, which can effectively protect the image. Notably, (e) is exactly the same as (b), and (f) is exactly the same as (a), demonstrating that our scheme is completely reversible.

Similar to Figs. 12 and Fig. 13 shows an example of our experiment with $(t, n) = (4, 4)$. We can see that the histograms of our encrypted image and share are still very smooth, and the noise image exhibits strong randomness. This demonstrates that it has good security performance under different settings and is not affected by the (t, n)-threshold.

Table 1 intuitively shows the performance of our scheme, in which the Number of Change Pixel Rate (NPCR) and the Unified Averaged Changed Intensity (UACI) are commonly used to evaluate image encryption algorithms. NPCR quantifies the proportion of pixel alterations between two images I_1 and I_2, as illustrated in Eq. (10) and Eq. (11), where i and j represent the coordinates of the pixels, D signifies the count of pixel changes and T represents the total number of pixels. In simple terms, a higher NPCR indicates a greater number of pixels that have undergone changes. It reflects the extent of change to the image during the encryption process. UACI can measure the average contrast intensity of pixel changes, as depicted in Eq. (12), where the numerator signifies the absolute value of the difference in pixel values between the two images. F in the denominator represents the maximum pixel value, which is 255, and T denotes the total number of pixels. UACI primarily reflects the average

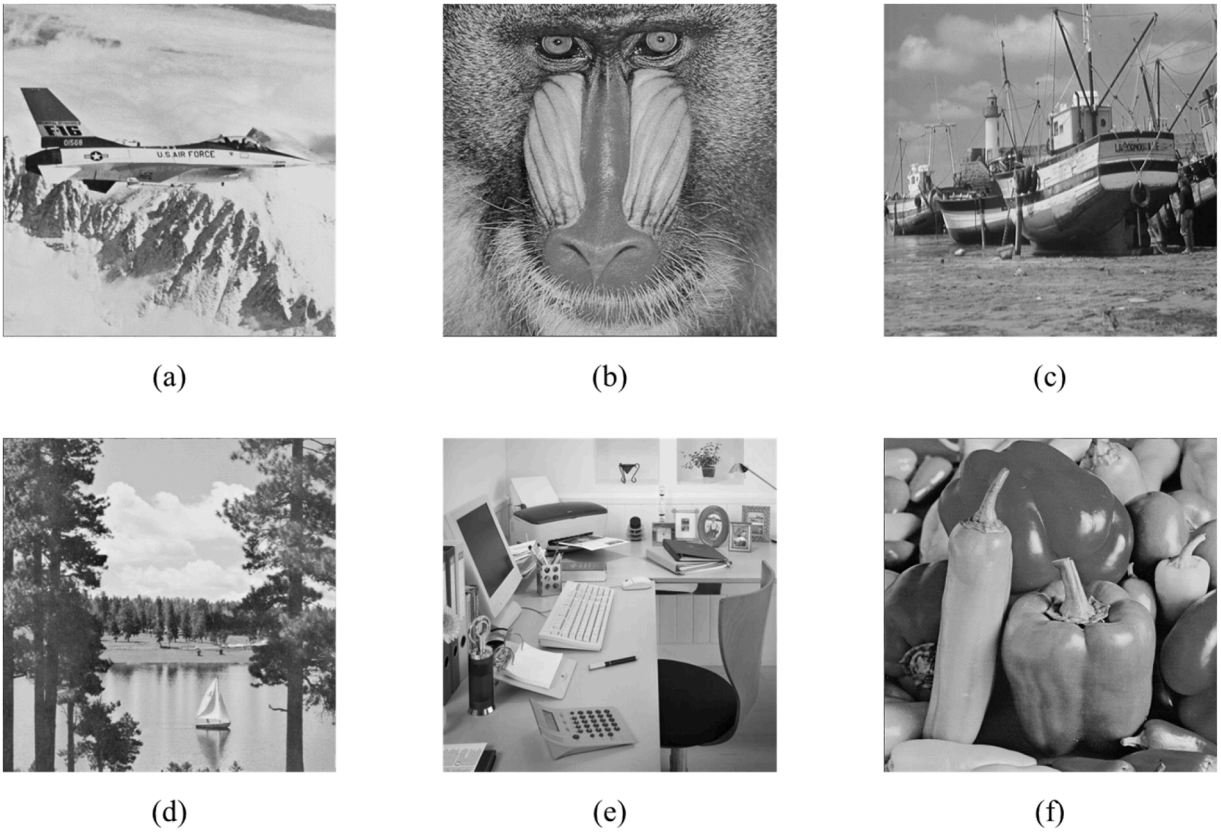

Fig. 11. Six test images: (a) Airplane; (b) Baboon; (c) Boat; (d) Lake; (e) Office; (f) Peppers.

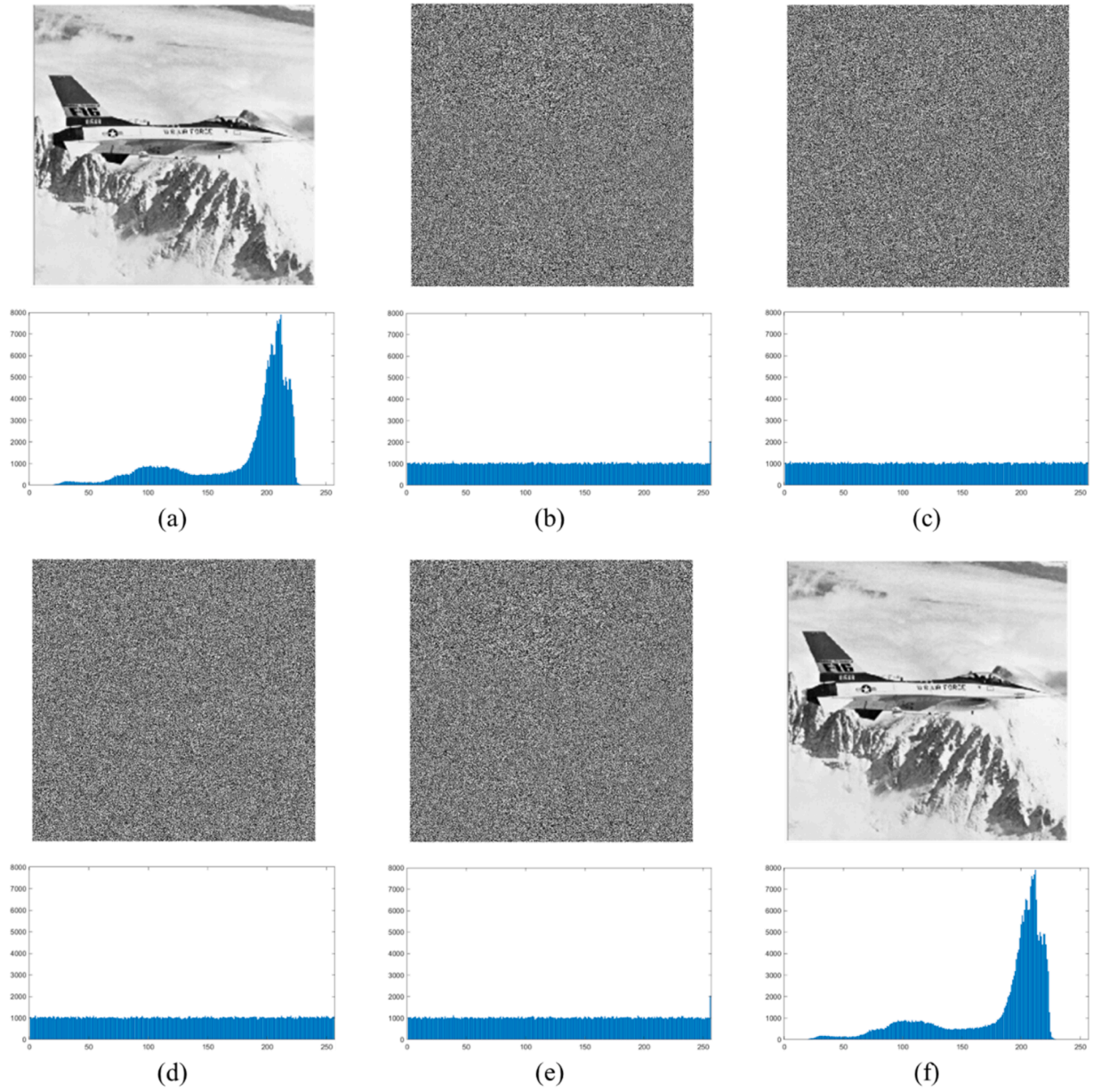

Fig. 12. Images and their histograms of our proposed scheme at different phases when $(t, n) = (2, 2)$: (a) Original image; (b) Encrypted image; (c)-(d) Shares; (e) Reconstructed encrypted image; (f) Reconstructed original image.

degree of change in the disparity between the original digital image data and the encrypted image data. The expectation for NPCR is 99.6094 %, and for UACI, it is 33.4635 % [39]. When the calculated result of the encryption algorithm closely matches these expectations, it indicates better encryption performance. In the results presented in Table 1, the average NPCR offered with our proposed scheme is approximately 99.60 %, and the average UACI is approximately 29.86 % for our proposed scheme, closely aligning with expectations. This demonstrates the strong encryption performance of our scheme.

$$NPCR = \sum_{i,j} \frac{D(i,j)}{T} \times 100\% . \tag{10}$$

$$D(i,j) = \begin{cases} 0, & I_1(i,j) = I_2(i,j) \\ 1, & I_1(i,j) \neq I_2(i,j) \end{cases} . \tag{11}$$

$$UACI = \sum_{i,j} \frac{|I_1(i,j) - I_2(i,j)|}{F \times T} \times 100\% . \tag{12}$$

In addition to evaluating encryption performance, we assessed visual quality using common metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [40]. As defined in Eq. (13) and Eq. (14), PSNR measures the visual quality difference between two images. Here, m and n represent the image's width and height, I denotes the original image, and K represents either the encrypted image or a share. A lower Mean Squared Error (MSE) indicates a higher similarity between images, as PSNR and MSE are inversely proportional. Generally, a PSNR above 30 dB is considered acceptable for good visual quality, while values below 20 dB indicate poor quality, with lower values corresponding to more significant degradation. In encrypted images, PSNR is typically low, reflecting the high distortion introduced by encryption. SSIM, defined in Eq. (15), quantifies the similarity between two images$x$ and y by evaluating luminance, contrast, and structure separately before combining them using weighted products. In this formula, μ_x and μ_y represent the average luminance, σ_x and σ_y denote the standard deviation of luminance, and σ_{xy} represents the covariance between x and y. A constant C is included for stability. An

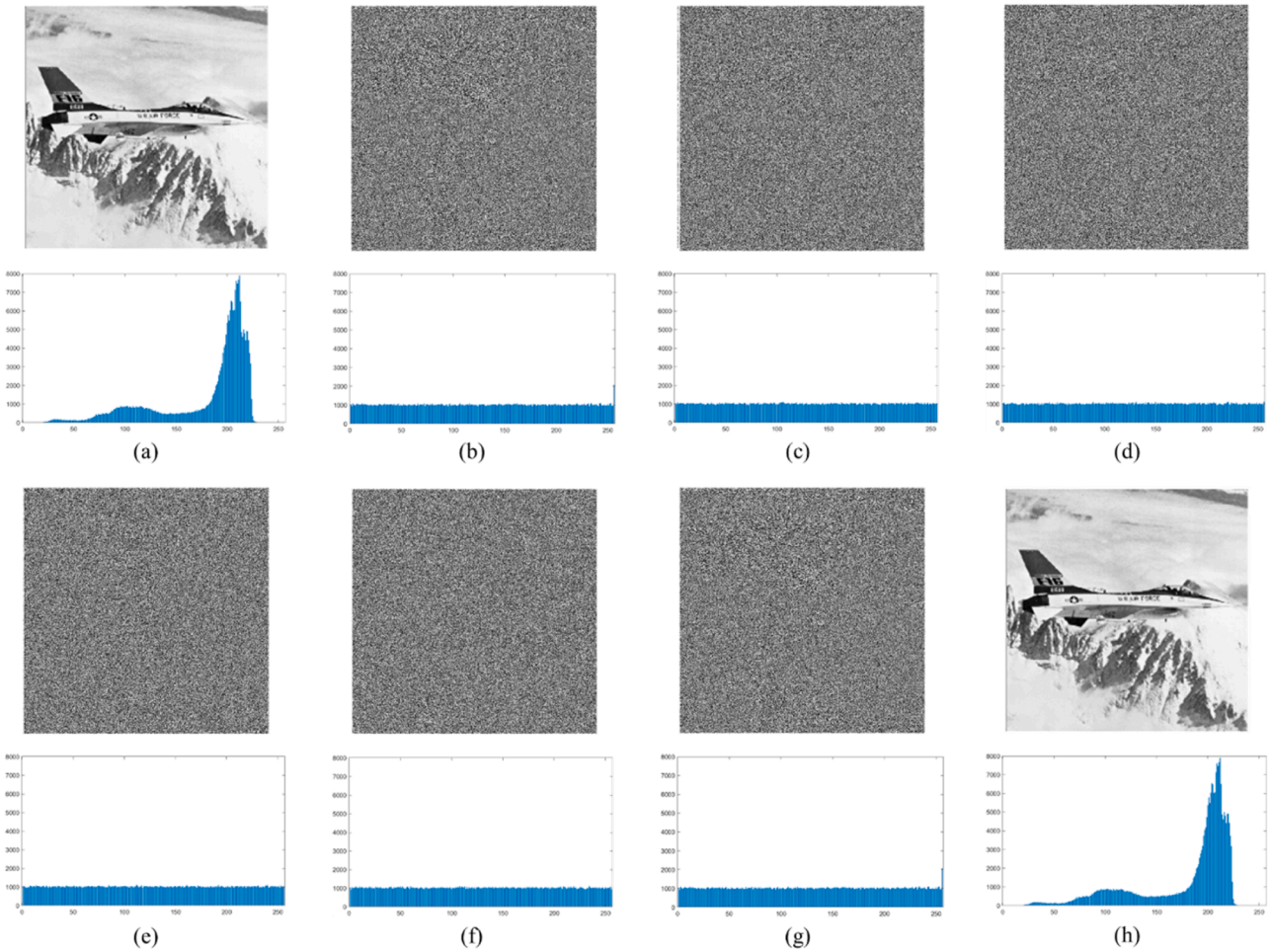

Fig. 13. Images and their histograms of our proposed scheme at different phases when $(t, n) = (4, 4)$: (a) Original image; (b) Encrypted image; (c)-(f) Shares; (g) Reconstructed encrypted image; (h) Reconstructed original image.

Table 1
Performance of images in our proposed scheme when $(t, n) = (2, 2)$.

Images		Airplane	Baboon	Boat	Lake	Office	Peppers	**Average**
NPCR (%)	Encrypted Image	99.6166	99.6113	99.6269	99.6056	99.6243	99.5979	99.6138
	Share 1	99.6094	99.6376	99.6025	99.6056	99.6052	99.6166	99.6128
	Share 2	99.6052	99.6075	99.6166	99.6101	99.6071	99.5918	99.6064
UACI (%)	Encrypted Image	32.3654	27.8705	28.4861	31.3900	31.3753	29.6766	30.1940
	Share 1	32.3519	27.86	28.4470	31.3952	31.2890	29.5734	30.1528
	Share 2	32.3519	27.8199	28.4873	31.3088	31.2779	29.5649	30.1351
PSNR (dB)	Encrypted Image	8.0370	9.5189	9.2844	8.3280	8.3302	8.8604	8.7265
	Share 1	8.0434	9.5227	9.2980	8.3245	8.3514	8.8908	8.7385
	Share 2	8.0380	9.5327	9.2848	8.3442	8.3496	8.8964	8.7410
SSIM	Encrypted Image	0.0099	0.0119	0.0093	0.0097	0.0093	0.0096	0.0100
	Share 1	0.0093	0.0108	0.0096	0.0096	0.0098	0.0101	0.0099
	Share 2	0.0101	0.0104	0.0093	0.0095	0.0099	0.0102	0.0099

SSIM value closer to 1 indicates greater similarity, while a value approaching 0 signifies higher divergence. For encrypted images, an ideal SSIM is as close to 0 as possible, ensuring that encryption introduces significant differences from the original image. As shown in Table 1, the average PSNR for the encrypted image, Share 1, and Share 2 is approximately 8.98, while the SSIM is around 0.01. These values indicate a significant reduction in visual quality compared to the original images, demonstrating the effectiveness of the encryption algorithm and share construction process.

$$PSNR = 10 \times log_{10}\frac{255^2}{MSE}. \tag{13}$$

$$MSE = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j) - K(i,j)]^2. \tag{14}$$

$$SSIM(x,y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}. \tag{15}$$

Table 2 shows the statistical analyses of our proposed scheme. Horizontal correlation and vertical correlation are used to describe the degree of correlation between adjacent pixels in the image in the corresponding direction. Eq. (16) shows the correlation calculation

method, where *Mean* represents the mean, and *Var* represents the variance. The larger the value of *Corr*, the higher the correlation between adjacent pixels, and the smaller the value, the lower the correlation. From Table 2, we can observe that both the horizontal and vertical correlations of the original image are very close to 1, indicating high correlation among adjacent pixels in the original image. For encrypted image and shares, the horizontal and vertical correlations is around 0, indicating almost no correlation between adjacent pixels. Information entropy is used to measure the complexity or amount of information in an image. Eq. (17) shows the entropy algorithm, where $p(x)$ represents the probability that a certain pixel value x appears in the image. The greater the entropy of the image, the more complex the surface image is. In other words, the closer the entropy of the encrypted image is to 8, the more uniform the distribution of pixel values in the image. As shown in Table 2, the average entropy of encrypted image and shares is approximately 7.99, indicating a high level of complexity and an even distribution. This observation aligns with our scheme's strong encryption performance. It is consistent with the conclusion that the histogram distribution is uniform.

$$Corr_{x,y} = \frac{Mean\{[x - Mean(x)] \times [y - Mean(y)]\}}{\sqrt{Var(x)Var(y)}} . \tag{16}$$

$$H(X) = -\sum [p(x) \times log_2(p(x))] . \tag{17}$$

To illustrate the superiority of the expandability of our proposed scheme, we present the embedding rates under different search square side lengths l in Table 3. As described in Eq. (18), bits per pixel (bpp) represents the average number of bits that can be embedded in each pixel, with its numerator being the total number of bits, known as the embedding capacity (EC). In Table 3, the embedding bits (EB) indicate the number of secret bits that can be embedded in each search square. Referencing Algorithm 2, we observe that EB is determined by $\lfloor log_2 l^2 \rfloor$, and thus, the same EB may occur at different l values. From the experimental results, it is evident that our scheme supports $l \in [2, 257]$. Specifically, when $l \in [256, 257]$, the maximum EB is 16 bits, corresponding to a maximum EC of 2097,152 bits. When $(t, n) = (2, 2)$, the maximum bpp is 4. This observation suggests that as the l grows, the embedding rate also increases. Consequently, we have the flexibility to adjust the l value according to security and embedding capacity needs. This adaptability and performance improvement are distinctive features of our proposed scheme.

$$bpp = \frac{Total\ Number\ of\ Bits}{Total\ Number\ of\ Pixels} . \tag{18}$$

Table 4 shows the comparison with other RDHEI schemes. Except for the scheme in [31], which is irreversible to the original image, all other schemes can achieve lossless recovery. Except for the schemes in [31] and [22], other schemes are separable. Except for the schemes in [22,32] and one scheme in [23], which require preprocessing before encryption, none of the other methods requires additional preprocessing.

Table 3
Embedding rates at different l when $(t, n) = (2, 2)$.

l	EB	EC	bpp
2	2	262,144	0.5
3	3	393,216	0.75
[4,5]	4	524,288	1
[6,7]	5	655,360	1.25
[8,11]	6	786,432	1.5
[12,15]	7	917,504	1.75
[16,22]	8	1048,576	2
[23,31]	9	1179,648	2.25
[32,45]	10	1310,720	2.5
[46,63]	11	1441,792	2.75
[64,90]	12	1572,864	3
[91,127]	13	1703,936	3.25
[128,181]	14	1835,008	3.5
[182,255]	15	1966,080	3.75
[256,257]	16	2097,152	4

Preprocessing can often be a time-consuming and computationally intensive task, so schemes that avoid it are generally more efficient. Furthermore, the methodologies discussed in references [23,30,31–33], including our proposed approach, are distinguished by their robust security measures. References [35,36] utilize homomorphic encryption, enabling the attainment of even higher security standards. Similarly, [23,32,33], along with our proposed method, use a secret sharing mechanism to ensure the confidentiality and security of image data. Notably, the techniques outlined in [23,32,33], alongside our proposal, are resilient against single points of failure due to their integration of secret sharing and RDHEI. Conversely, references [22,26–31] employ a secret sharing scheme but encrypt the image into a solitary encrypted image, making it vulnerable to a single point of failure. In the event that this sole encrypted image is compromised or lost, image recovery becomes unfeasible.

Table 5 presents a comparison between our SS-based RDHEI schemes and those proposed by other researchers, except for the scheme proposed by Chen et al. [22], all others employ (t, n) secret sharing schemes. Obviously, all schemes can prevent data leakage from CSPs and prevent malicious collusion between CSPs. As delineated in the Introduction Section, it's important to note that in other schemes, the data hiders are typically cloud servers. This means that external entities, rather than the content owners themselves, embed supplementary information such as ownership details of the cloud servers via multi-data hiders. Conversely, our scheme supports content owners in asserting ownership claims. Due to the use of the reference matrix, our scheme designs appropriate data embedding strategies to enhance both the capacity and efficiency of information hiding. Both the average and maximum payloads of our scheme reach up to 4 bpp, outperforming other SS-based RDHEI schemes. Compared to Hua et al.'s scheme [38], our scheme offers a maximum payload that is 0.49 bpp greater.

We also conducted experiments using images extracted from two gray standard image datasets: BOSSbase [41], from which we used 100

Table 2
Statistical analysis of images in our proposed scheme when $(t, n) = (2, 2)$.

Image		Airplane	Baboon	Boat	Lake	Office	Peppers	**Average**
Horizontal Correlation	Original Image	0.9598	0.8653	0.9368	0.9739	0.9762	0.9733	0.9476
	Encrypted Image	0.0140	0.0217	−0.0412	0.0127	0.0011	−0.0246	−0.0027
	Share 1	−0.0043	0.0102	−0.0195	0.0088	−0.0016	−0.0084	−0.0025
	Share 2	−0.0034	−0.0007	−0.0021	0.0014	−0.0026	−0.0012	−0.0014
Vertical Correlation	Original Image	0.9613	0.7523	0.9709	0.9712	0.9719	0.9763	0.9340
	Encrypted Image	0.0011	−0.0023	−0.0038	0.0008	0.0001	−0.0036	−0.0013
	Share 1	−0.0020	0.0028	−0.0010	−0.0015	0.0025	0.0022	0.0005
	Share 2	−0.0038	−0.0002	−0.0010	−0.0028	0.0014	0.0003	−0.0010
Entropy	Original Image	6.7059	7.3579	7.1914	7.4570	7.6050	7.5944	7.3186
	Encrypted Image	7.9973	7.9968	7.9973	7.9972	7.9973	7.9968	7.9971
	Share 1	7.9992	7.9993	7.9993	7.9992	7.9994	7.9994	7.9993
	Share 2	7.9994	7.9993	7.9994	7.9994	7.9993	7.9993	7.9994

Table 4
Comparison with other RDHEI schemes.

Schemes	Encryption Strategies	Lossless Recovery	Separable	Free-Preprocessing	Security	Resistance to Single Point of Failure
Bhardwaj et al. [31]	XOR	×	×	√	Limited	×
Yi et al. [32]	Block permutation and co-modulation	√	√	√	Limited	×
Liu et al. [33]	In-block pixel permutation and co-XOR	√	√	√	Limited	×
Wang et al. [34]	Block permutation and co-XOR	√	√	√	Limited	×
Zheng et al. [35]	Homomorphic encryption	√	√	√	High	×
Chen et al. [36]	Homomorphic encryption	√	√	√	High	×
Chen et al. [22]	Secret sharing	√	×	×	Limited	×
Chen et al. [37]	Secret sharing	√	√	×	H igh	√
Qin et al. [23]	Secret sharing	√	√	√/ ×	High	√
Hua et al. [38]	Secret sharing	√	√	√	High	√
Proposed Scheme	Secret sharing	√	√	√	High	√

images, and BOW-2 [42], also with 100 images, as well as a medical image dataset, Osirix [43], from which we used 50 images. Table 6 shows the performance of different datasets in our proposed scheme when $(t, n) = (2, 2)$. We can observe that for the encrypted images on all datasets, share 1, share 2, and NPCR are around 99.60 %, and SSIM is close to 0. On the gray standard image datasets BOSSbase and BOW-2, UACI is around 0.3 %, and PSNR is around 8 dB. On the medical image dataset Osirix, UACI is around 0.4 %, and PSNR is around 6 dB. The main reason for the difference with the gray standard image datasets is that medical images usually have a large number of black background areas. Table 7 shows the statistical analysis of different datasets in our proposed scheme when $(t, n) = (2, 2)$. We can find that the horizontal correlation and vertical correlation of encrypted images, Share 1, and Share 2 are very close to 0, and their entropies are very close to 8, which means that the shares generated by our proposed scheme meet the requirements of encrypted images.

Data expansion in RDHEI schemes is observed when the size of the marked encrypted image exceeds that of the original image. This expansion is quantified by the expansion rate, defined as the ratio of the total bits in the marked encrypted image to those in the original image. We analyze this phenomenon in RDHEI schemes that utilize homomorphic encryption and secret sharing, both of which provide enhanced security. Table 8 illustrates a comparison of data expansion between these schemes. Homomorphic encryption-based RDHEI schemes [35,36], which support user-set security parameters, experience significant data expansion. Conversely, secret sharing-based RDHEI schemes encrypt the original image into n encrypted images and distribute them to n data hiders for embedding. The secret sharing-based schemes described in [22,19,32,33], as well as our proposed scheme, demonstrate substantially less data expansion with acceptable expansion rates. For these schemes, while the overall expansion rate is n, each individual data hider experiences an expansion rate of 1, indicating minimal data expansion, as discussed in [38] and in our analysis.

Table 9 illustrates the security analysis of our proposed scheme, detailing the corresponding scenarios and results for three key measurements. Their definitions are as follows: (a) Confidentiality ensures that when fewer than t shares are available, the secret remains completely unrecoverable, with no partial leakage of information. Our scheme upholds this theoretical guarantee—an attacker with fewer than t shares cannot retrieve any part of the secret, thereby maintaining full confidentiality. (b) Tampering resistance ensures that any modification to the shares prevents successful secret reconstruction. In our approach, if shares are tampered with, the reconstruction process is corrupted, rendering the altered portions irrecoverable and preserving the integrity of the secret. (c) Collusion resistance guarantees that even if t or more participants conspire, they cannot reconstruct the secret unless they possess the decryption key or additional critical information. In our method, even if t cloud servers collude, they can only reconstruct the encrypted image. Without the decryption key, they remain unable to access the original secret, ensuring strong protection against coordinated attacks.

To further illustrate these security properties, Fig. 14 provides visualization results, with each row representing a different example. In the first group, Fig. 14(a) displays Share 1 of the "Airplane" with a tampered black region, while Fig. 14(b) shows an unaltered Share 2 of the "Airplane." The reconstructed image in Fig. 14(c) reveals that only the tampered area remains unrecoverable, while the rest is restored without loss, demonstrating the robustness of our scheme. In the second group, Fig. 14(d) depicts Share 1 of the "Baboon." Unlike the previous case, Fig. 14(e) has been altered with noise, making the modifications visually undetectable, as indicated by the red line. However, in the reconstructed image, Fig. 14(f), only the tampered region marked by the red line remains unrecoverable, demonstrating the scheme's robustness against tampering. The third group examines the effect of using unrelated shares. Fig. 14(g) represents Share 1 of the "Airplane," while Fig. 14(h) corresponds to Share 2 of the "Baboon." As a result, the reconstruction process fails entirely, highlighting the strong confidentiality of the proposed scheme.

Table 10 presents a comparison of the theoretical time complexities of various secret sharing techniques, where MN denotes the image size. For the scheme proposed by Chen et al. [22] and our proposed method, the addition complexity is $(2t - 1)MN$, while the multiplication complexity is $t(2t - 1)MN$. Conversely, the schemes by Chen et al. [37] and Qin et al. [23] share identical complexities, with addition

Table 5
Comparison with other SS-based RDHEI schemes.

Scheme Criterion	Chen et al. [22]	Chen et al. [37]	Qin et al. [23]	Hua et al. [38]	Proposed Scheme
(t, n)Secret Sharing	Extension version of (t, n) SS as $(k - d, d, k, n)$ where $d = t$, $k = 2t$ and $n = 2t$	√	√	√	√
Prevent the Leakage of Data from CSPs	√	√	√	√	√
Prevent Malicious Collusion among CSPs	√	√	√	√	√
Support Content Owners in Asserting Ownership Claims	×	×	×	×	√
Average Payload (bpp)	0.498, fixed payload	3.5, fixed payload	1.289	2.054	4, fixed payload
Maximum Payload (bpp)	0.498	3.5	1.581	3.510	4

Table 6
Performance of different datasets in our proposed scheme when $(t, n) = (2, 2)$.

Images		BOSSbase			BOWS-2			Osirix		
		Maximum	Minimum	Average	Maximum	Minimum	Average	Maximum	Minimum	Average
NPCR (%)	Encrypted Image	99.6887	99.5041	99.6027	99.6380	99.5602	99.6085	99.6619	99.5625	99.6125
	Share 1	99.6643	99.5667	99.6084	99.6449	99.5731	99.6093	99.6550	99.5850	99.6188
	Share 2	99.6490	99.5514	99.6113	99.6525	99.5617	99.6077	99.6450	99.5725	99.6096
UACI (%)	Encrypted Image	0.4203	0.2562	0.3343	0.3741	0.2682	0.3071	0.4865	0.2781	0.4044
	Share 1	0.4172	0.2556	0.3330	0.3729	0.2673	0.3065	0.4835	0.2771	0.4029
	Share 2	0.4190	0.2565	0.3331	0.3731	0.2676	0.3065	0.4818	0.2766	0.4024
PSNR (dB)	Encrypted Image	10.4796	5.9457	7.8729	9.9565	6.8244	8.5805	9.5465	4.9314	6.3061
	Share 1	10.5150	5.9967	7.9067	9.9798	6.8566	8.5991	9.5752	4.9894	6.3397
	Share 2	10.4895	5.9760	7.9058	9.9718	6.8499	8.5986	9.5885	5.0072	6.3476
SSIM	Encrypted Image	0.0115	0.0032	0.0080	0.0114	0.0068	0.0096	0.0099	−0.0003	0.0047
	Share 1	0.0117	0.0012	0.0083	0.0116	0.0062	0.0097	0.0099	−0.0005	0.0046
	Share 2	0.0113	0.0041	0.0083	0.0118	0.0060	0.0097	0.0097	0.0002	0.0046

Table 7
Statistical analysis of different datasets in our proposed scheme when $(t, n) = (2, 2)$.

Image		BOSSbase			BOWS-2			Osirix		
		Maximum	Minimum	Average	Maximum	Minimum	Average	Maximum	Minimum	Average
Horizontal Correlation	Original Image	0.9958	0.7865	0.9519	0.9974	0.8461	0.9717	0.9970	0.8876	0.9605
	Encrypted Image	0.1399	−0.0739	−0.0005	0.2112	−0.1751	−0.0001	0.4328	−0.1463	0.2001
	Share 1	0.0570	−0.0292	−0.0009	0.0555	−0.0660	0.0006	0.1220	−0.0523	0.0543
	Share 2	0.0107	−0.0074	0.0003	0.0082	−0.0059	0.0003	0.0073	−0.0162	−0.0057
Vertical Correlation	Original Image	0.9864	0.8041	0.9442	0.9932	0.7594	0.9676	0.9963	0.9185	0.9712
	Encrypted Image	0.0127	−0.0062	0.0006	0.0058	−0.0070	0.0001	0.0071	−0.0054	0.0006
	Share 1	0.0063	−0.0099	−0.0002	0.0056	−0.0052	0.0000	0.0065	−0.0044	−0.0003
	Share 2	0.0066	−0.0072	−0.0011	0.0061	−0.0059	0.0000	0.0036	−0.0085	−0.0008
Entropy	Original Image	7.7565	4.7257	7.0063	7.8788	5.5026	7.0695	7.3880	1.8095	5.3065
	Encrypted Image	7.9959	7.9940	7.9950	7.9977	7.9966	7.9971	7.9971	7.9956	7.9963
	Share 1	7.9976	7.9963	7.9972	7.9995	7.9990	7.9993	7.9990	7.9917	7.9970
	Share 2	7.9977	7.9959	7.9971	7.9994	7.9987	7.9993	7.9989	7.9921	7.9969

Table 8
Comparison of data expansion rates with other RDHEI schemes.

Schemes	Content Owner	Each Data Hider
Zheng et al. [35]	8, 16	8, 16
Chen et al. [36]	64, 128	64, 128
Chen et al. [22]	n	1
Chen et al. [37]	n	1
Qin et al. [23]	n	1
Hua et al. [38]	n	1
Proposed Scheme	n	1

Table 9
Security analysis of our proposed scheme.

Metric	Setup	Result
Confidentiality	Select fewer than t shares and attempt to recover the secret.	Fewer than t shares cannot recover the secret.
Tampering Resistance	Modify one or more shares and attempt to recover the secret.	Tampered shares do not lead to correct recovery of the secret.
Collusion Resistance	Simulate malicious participants collaborating to recover the secret.	Malicious participants cannot recover the secret without the key.

complexity of $(t-1)MN$ and multiplication complexity of $\frac{t(t-1)}{2}MN$. The scheme by Hua et al. [38] introduces additional complexity based on the block size S, where the addition complexity is $\frac{(2t-4+S^2)}{S^2}MN$ and the multiplication complexity is $\frac{(2t-3)}{S^2}MN$. It is worth noting that these schemes illustrate the inherent trade-offs and limitations of classical (t, n) secret sharing. Notably, achieving strong security often incurs increased computational overhead and longer processing times—necessary compromises for ensuring robust protection. These design choices highlight the ongoing challenge of balancing security and efficiency across different application scenarios. While the proposed schemes may not offer optimal theoretical time complexity, they provide greater embedding capacity than existing methods while maintaining strong security performance.

6. Conclusions

This paper introduces a novel RDHEI scheme that combines (t, n) secret sharing with encryption and expandable magic matrix-based data hiding. Our scheme provides strong resistance to hacker attacks, ensuring the security of IoT and cloud services. Unlike traditional approaches, our method enables the data owner, such as a hospital, to embed ownership data and assert ownership before uploading it to cloud platforms managed by CSPs, rather than relying on CSPs to embed the data at the end. In the experimental section, we demonstrate the superiority of our scheme by showcasing excellent encryption performance and the full reversibility of the process. By adjusting the side length of the search square, we can significantly improve the embedding rate, enabling the embedding of up to 2097,152 bits at maximum. Specifically, with $(t, n) = (2, 2)$, the bits per pixel can reach 4.

Notably, our approach enables content owners to directly embed ownership data, eliminating the need for cloud service providers (CSPs) to handle this step. Images are encrypted at the IoT gateway before upload, ensuring that even CSPs cannot access the original content. The encrypted shares are then distributed across multiple cloud servers, enhancing resistance to single-point attacks. This design ensures that only authorized users can reconstruct the original image, strengthening data security and protecting against unauthorized access or external threats. Our scheme not only supports (t,n) secret sharing but also delivers robust security and high embedding capacity, allowing content

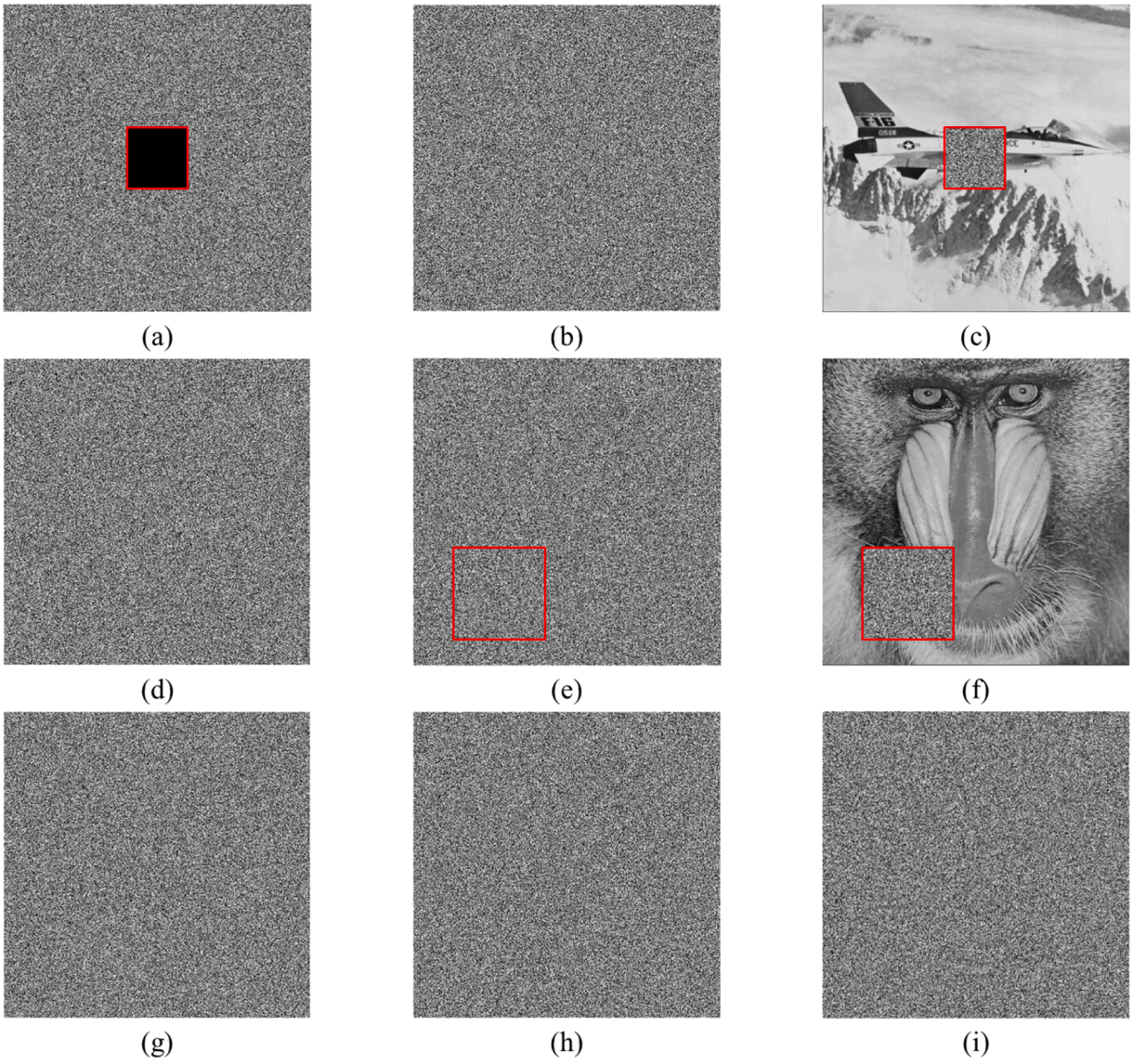

Fig. 14. Security analysis: (a) Share 1 of "Airplane" tampered with black; (b) Share 2 of "Airplane" not tampered with; (c) Reconstructed image decrypted using (a) and (b); (d) Share 1 of "Baboon" not tampered with; (e) Share 2 of "Baboon" tampered with noise; (f) Reconstructed image decrypted using (d) and (e); (g) Share 1 of "Airplane" not tampered with; (h) Share 2 of "Baboon" not tampered with; (i) Reconstructed image decrypted using (g) and (h).

Table 10
Theoretical time complexity of different secret sharing techniques.

Schemes	Addition	Multiplication
Chen et al. [22]	$(2t-1)MN$	$t(2t-1)MN$
Chen et al. [37]	$(t-1)MN$	$\frac{t(t-1)}{2}MN$
Qin et al. [23]	$(t-1)MN$	$\frac{t(t-1)}{2}MN$
Hua et al. [38]	$\frac{(2t-4+S^2)}{S^2}MN$	$\frac{(2t-3)}{S^2}MN$
Proposed Scheme	$(2t-1)MN$	$t(2t-1)MN$

owners to assert ownership with confidence. The trade-off for this increased capacity is higher computational overhead—a common challenge in secure system design. Future work will explore the scalability of the scheme in multi-party computation cloud environments, aiming to improve both security and efficiency across a broader range of practical applications.

Funding

This work was supported in part by the National Science and Technology Council under the grant NSTC 113–2410-H-167 −012 -MY3 and NSTC 113–2634-F-005 −001 -MBK.

CRediT authorship contribution statement

Yijie Lin: Writing – review & editing, Writing – original draft, Visualization, Software, Formal analysis, Data curation. **Chia-Chen Lin:** Writing – review & editing, Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Ching-Chun Chang:** Investigation, Formal analysis. **Chin-Chen Chang:** Validation, Supervision, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Chia-Chen Lin reports financial support was provided by National Science and Technology Council.

Data availability

Data will be made available on request.

References

[1] Tian J. Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 2003;13(8):890–6.
[2] Celik MU, Sharma G, Tekalp AM, Saber E. Lossless generalized-LSB data embedding. IEEE Trans Image Process 2005;14(2):253–66.
[3] Ni Z, Shi YQ, Ansari N, Su W. Reversible data hiding. IEEE Trans Circuits Syst Video Technol 2006;16(3):354–62.
[4] Thodi DM, Rodríguez JJ. Expansion embedding techniques for reversible watermarking. IEEE Trans Image Process 2007;16(3):721–30.
[5] Liu JC, Chang CC, Lin Y, Chang CC, Horng JH. A matrix coding-oriented reversible data hiding scheme using dual digital images. Mathematics 2023;12(1):86.
[6] Lin Y, Liu JC, Chang CC, Chang CC. Embedding secret data in a vector quantization codebook using a novel thresholding scheme. Mathematics 2024;12(9):1332.
[7] Rivest RL. Cryptography. Algorithms and complexity. Elsevier; 1990. p. 717–55.
[8] Zhang X. Reversible data hiding in encrypted image. IEEE Signal Process Lett 2011; 18(4):255–8.
[9] Zhang X. Separable reversible data hiding in encrypted image. IEEE Trans Inf Forensics Secur 2011;7(2):826–32.
[10] Hong W, Chen TS, Wu HY. An improved reversible data hiding in encrypted images using side match. IEEE Signal Process Lett 2012;19(4):199–202.
[11] Ma K, Zhang W, Zhao X, Yu N, Li F. Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inf Forensics Secur 2013;8(3): 553–62.
[12] Wu X, Sun W. High-capacity reversible data hiding in encrypted images by prediction error. Signal Process 2014;104:387–400.
[13] Lin Y, Lin CC, Chang CC, Chang CC. An IoT-based electronic health protection mechanism with AMBTC compressed images. IEEE Internet of Things J 2024.
[14] Zhang X, Zhang X, Yu C, Yang J, Tang Z. A multi-directional gradient predictor for region-based reversible data hiding in encrypted images. IEEE Int Things J 2024.
[15] Wu X, Weng J, Yan W. Adopting secret sharing for reversible data hiding in encrypted images. Signal Process 2018;143:269–81.
[16] Shamir A. How to share a secret. Commun ACM 1979;22(11):612–3.
[17] Thien CC, Lin JC. Secret image sharing. Comput Graph 2002;26(5):765–70.
[18] Xiong L, Ding R, Yang CN, Fu Z. Invertible secret image sharing with authentication for embedding color palette image into true color image. IEEE Trans Circuits Syst Video Technol 2024.
[19] Wang R, Li L, Yang G, Yan X, Yan W. Secret cracking and security enhancement for the image application of CRT-based Secret sharing. IEEE Trans Inf Forensics Secur 2024.
[20] Jiang Y, Chen K, Yan W, Yan X, Yang G, Zeng K. Robust secret image sharing resistant to JPEG recompression based on stable block condition. IEEE Trans Multimedia 2024.
[21] Xiong L, Ding R, Yang CN, Fu Z. Robust secret image sharing scheme based on polynomial k-consistency. IEEE Trans Circuits Syst Video Technol 2025.
[22] Chen YC, Hung TH, Hsieh SH, Shiu CW. A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms. IEEE Trans Inf Forensics Secur 2019;14(12):3332–43.
[23] Qin C, Jiang C, Mo Q, Yao H, Chang CC. Reversible data hiding in encrypted image via secret sharing based on GF (p) and GF (2^8). IEEE Trans Circuits Syst Video Technol 2021;32(4):1928–41.
[24] Hua Z, Wang Y, Yi S, Zheng Y, Liu X, Chen Y, Zhang X. Matrix-based secret sharing for reversible data hiding in encrypted images. IEEE Trans Dependable Secure Comput 2022;20(5):3669–86.
[25] Zhang X, Wang S. Efficient steganographic embedding by exploiting modification direction. IEEE Commun Lett 2006;10(11):781–3.
[26] Kim HJ, Kim C, Choi Y, Wang S, Zhang X. Improved modification direction methods. Comput Math Appl 2010;60(2):319–25.
[27] Kieu D, Wang ZH, Chang CC, Li MC. A Sudoku based wet paper hiding scheme. Int J Smart Home 2009;3(2):1–12.
[28] Chang CC, Liu Y, Nguyen TS. A novel turtle shell based scheme for data hiding. In: 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing. IEEE; 2014, August. p. 89–93.
[29] Lin Y, Lin CC, Liu JC, Chang CC. Verifiable (t,n) secret image sharing scheme based on slim turtle shell matrix. J InfSecurity Applications 2024;80:103679.
[30] Lin Y, Liu JC, Chang CC, Chang CC. A puzzle matrix oriented secret sharing scheme for dual images with reversibility. Signal Process 2025:110056.
[31] Bhardwaj R, Aggarwal A. An improved block based joint reversible data hiding in encrypted images by symmetric cryptosystem. Pattern Recognit Lett 2020;139: 60–8.
[32] Yi S, Zhou Y. Separable and reversible data hiding in encrypted images using parametric binary tree labeling. IEEE Trans Multimedia 2018;21(1):51–64.
[33] Liu ZL, Pun CM. Reversible data hiding in encrypted images using chunk encryption and redundancy matrix representation. IEEE Trans Dependable Secure Comput 2020;19(2):1382–94.
[34] Wang Y, He W. High capacity reversible data hiding in encrypted image based on adaptive MSB prediction. IEEE Trans Multimedia 2021;24:1288–98.
[35] Zheng S, Wang Y, Hu D. Lossless data hiding based on homomorphic cryptosystem. IEEE Trans Dependable Secure Comput 2019;18(2):692–705.
[36] Chen B, Wu X, Lu W, Ren H. Reversible data hiding in encrypted images with additive and multiplicative public-key homomorphism. Signal Process 2019;164: 48–57.
[37] Chen B, Lu W, Huang J, Weng J, Zhou Y. Secret sharing based reversible data hiding in encrypted images with multiple data-hiders. IEEE Trans Dependable Secure Comput 2020;19(2):978–91.
[38] Hua Z, Liu X, Zheng Y, Yi S, Zhang Y. Reversible data hiding over encrypted images via preprocessing-free matrix secret sharing. IEEE Trans Circuits Syst Video Technol 2023.
[39] Wu Y, Noonan JP, Agaian S. NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J Selected Areas Telecomm 2011;1(2):31–8.
[40] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.
[41] Bas P, Filler T, Pevný T. Break our steganographic system": the ins and outs of organizing BOSS. International workshop on information hiding. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011, May. p. 59–70.
[42] Bas P, Furon T. Image database of BOWS-2. http://bows2.ec-lille.fr/; 2017.
[43] Osirix Database. [Online]. Available: https://www.osirix-viewer.com/support/knowledge-base/. [Accessed 21 July 2024].