A survey of reversible data hiding in encrypted images – The first 12 years

Pauline Puteaux, Simying Ong, Koksheik Wong, William Puech

Journal of Visual Communication and Image Representation (2021)

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

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

A Survey of Reversible Data Hiding in Encrypted Images - The First 12 Years

Pauline Puteaux[a], SimYing Ong[b], KokSheik Wong[c], William Puech[a,*]

[a] *LIRMM – Univ. Montpellier / CNRS, 860 rue de Saint Priest, 34095 Montpellier Cedex 5, France*
[b] *Faculty of Computer Science and Information Technology, University of Malaya, Malaysia*
[c] *School of Information Technology, Monash University Malaysia, Malaysia*

ARTICLE INFO

Article history:

Keywords: Multimedia security, image encryption, data hiding, signal processing in the encrypted domain.

ABSTRACT

In the last few years, with the increasing popularity of cloud computing and the availability of mobile smart devices as well as ubiquitous network connections, more and more users are uploading their personal data to remote servers. However, this can lead to significant security breaches, where confidentiality, integrity and authentication are constantly threatened. To overcome these multiple problems, multimedia data must be secured, for example by means of encryption before transmission and storage. In this survey, we look into the issues involved in handling encrypted multimedia data, and more specifically we focus on reversible data hiding in encrypted images (RDHEI). The aim of this survey is to present the birth and evolution of RDHEI methods over the last 12 years. We first highlight different classes and characteristics of RDHEI, then describe representative RDHEI methods. A comparison table is presented to summarize the key features and achievements of each representative RDHEI method considered in this survey. Finally, we share the future outlook of emerging applications and open research topics relevant to RDHEI for the next 12 years and beyond.

1. Introduction

In the age of omnipresent network connectivity and the availability of powerful general computing resources at affordable prices, the digital image is certainly worth more than a thousand words. To put statistics into context, more than 147,000 pictures are posted on Facebook alone every 60 seconds [1]. Furthermore, many email communications contain images as attachments and images are also sent via instant messaging platforms. Moreover, people often express themselves by using emoticons and short animated Graphics Interchange Format (aGIF) images, which add dynamism to the conventional text-based / message communication.

While the image has certainly revolutionized our daily lives, it is subject to various threats. Some of the classical examples include pictures of public figures (*e.g.* politicians) being edited to favor a particular group, pictures of celebrities' private lives being

*Corresponding author: Tel.: +33467418685; fax: +33467418500;
e-mail: pauline.puteaux@lirmm.fr (Pauline Puteaux), simying.ong@um.edu.my (SimYing Ong), wong.koksheik@monash.edu (KokSheik Wong), william.puech@lirmm.fr (William Puech)

exposed to the public, copyrighted images (including books and documents) being copied then sold at lower prices, to name a few.

Foreseeing these problems, researchers have proposed various techniques to protect digital images in general. One of the commonly researched approaches is called data hiding, where a data (payload) is embedded into a host image to serve a specific purpose. Here, the data content can be independent from the image (*i.e.* external), derived / extracted from the image, or a combination of both [2]. One of the most extensively researched areas is *watermarking*, where the owner's copyright data is embedded into the image [3]. When there is any dispute over the unauthorized utilization of a watermarked image, the embedded data can be extracted to prove ownership. The embedded watermark data is designed to withstand various forms of attack. In other words, the watermarked image is rendered useless (*i.e.* with most details being destroyed) when the embedded watermark fails to be recovered from the processed image, essentially a lose-lose situation. Another application is *fragile watermarking*, where the embedded watermark serves as a verification code, such as an authenticity or integrity check [4]. When any part of the watermark-protected image show any discrepancies, those parts are subjected to investigation, *i.e.* could they be forged or tampered with. To cater for the scenario where the host image cannot afford any form of distortion (*e.g.* medical, military, rare artwork), reversible techniques are proposed so that the embedded data can be removed to perfectly restore the original image [5, 6, 7, 8, 9, 10].

Encryption can also aim to conceal the perceptional meaning of an image. It aims to protect the image from unauthorized viewing, which promotes privacy and combats piracy. Almost all image encryption approaches rely on two basic building blocks, namely, permutation and substitution [11]. Some advanced image encryption techniques also make these operations dependent on the statistics / attributes of the input image, making cryptanalysis more challenging [12, 13]. In fact, in the case of images, the notion of security can be viewed from different perspectives. Specifically, in addition to the traditional cryptanalysis where the goal is to obtain the secret/private key, obtaining some idea of the plaintext image (*e.g.* outline) is sufficient [14]. In other words, the adversary does not need to get an exact copy of the original image, but a general outline may be sufficient. In any case, the practice of encrypting an image (as well as all other files) is becoming more common today due to online cloud storage services which are also available at no financial cost.

For many years, researchers in both fields have been making breakthroughs independently. However, both data hiding and encryption can be jointly deployed to provide extra features. One of the earliest efforts in combining data hiding and encryption has been proposed by Puech *et al.* in 2008 [15]. Researchers then started to invest in this direction, this is backed up by the increasing number of publications related to the unification of both data hiding and encryption. Fig. 1 depicts the number of publications regarding the joint deployment of data hiding and encryption in the last 12 years. Notably, the number of publications in 2019 is ~40 times more than that of 2009. This drastic increase is driven by the needs of various applications, including:

1. Digital rights management (buyer-seller protocol): Prior to transmission, a content seller encrypts the content to avoid unauthorized viewing. To deter illegal distribution by the buyer, the seller watermarks the content as well as embedding the buyer's information (*i.e.* fingerprinting), making each sold content a unique copy for tracking and tracing purposes. However, it is possible that the seller can *frame* an honest buyer by embedding his fingerprint into the content and distribute it for free. In order to prevent framing, the buyer-seller protocol is put forward so that the seller embeds a watermark, in which the final form of the watermark is not known to the seller [16]. A thorough treatment of encryption and fingerprinting in video can be found in [17].
2. Cloud storage: For privacy purposes, a user could encrypt his personal contents, including audio, image and video, prior to uploading them to a cloud for storage. For management purposes, the cloud administrator embeds data into the encrypted content, including data needed for indexing and statistics collection.This inspired the application of data hiding in encrypted

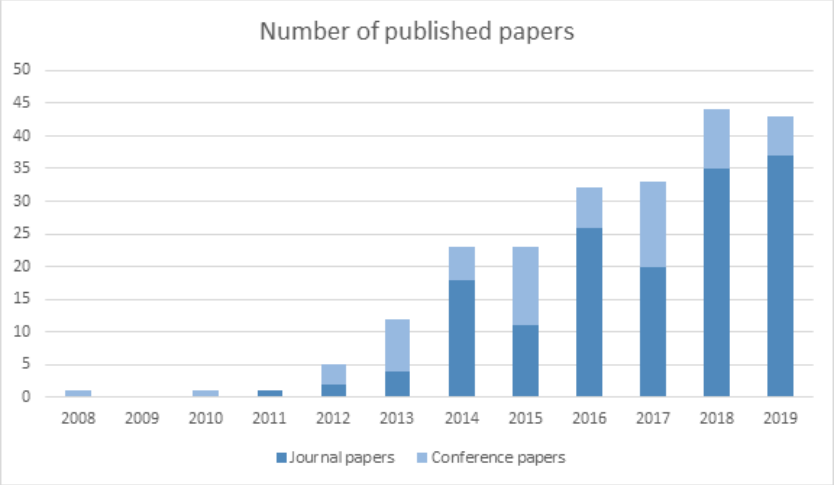

Fig. 1: History of the publications in RDHEI in the last 12 years.

content [18].

3. Patient's privacy: In hospitals, diagnosis images / documents of a patient are encrypted to ensure privacy, and patient's information is embedded into these contents for identification purposes. The nurses handling these contents can extract the embedded information from the encrypted content. The same information can be considered to link contents of a particular patient. On the other hand, the doctors / specialists can have access to both the patient's identification information as well as the original (*i.e.* plaintext) diagnosis images / documents.
4. Classified information: In a military situation, an under-officer with a lower clearance (*e.g.* clerk) can extract the embedded labels from the encrypted files (*e.g.* videos, documents, audio) to administer the files (*e.g.* copy, archive, move, *etc.*). On the other hand, the senior officer with a higher clearance (*e.g.* general) can access both embedded labels and the original files.
5. Reporter from the field: A reporter would encrypt an audio/image/video before transmitting it back to head quarters so that only the designated officers (especially not those from the rival news companies) can have access to the content for presenting exclusive coverage of the incident / event. Information such as the sender ID and Global Positioning System (GPS) location of the field reporter can be embedded for authentication purposes [19] to avoid content forgery (*e.g.* framing by rival companies).
6. Surveillance video recording: Video recording from a surveillance camera is selectively encrypted (*e.g.* region-of-interest masking) to avoid privacy infringement [20]. In addition, the camera ID, time, date, *etc.* are embedded because they are crucial information for authentication purposes, especially when the recording is presented during a court session as evidence.
7. Data analytic: Data is massively generated / collected nowadays thanks to the advent of data intensive science discovery [21]. The data needs to be labelled, but at the same time some form of privacy / secrecy must be ensured. The joint adaptation method can be readily deployed to handle such information.

In general, data hiding and encryption can be used together in 4 different ways, namely: 1) Insert-then-encrypt (ITE), 2) Encrypt-then-insert (ETI), 3) Insert-to-encrypt (I2E), and 4) Encrypt-to-insert (E2I) [22]. In each of these cases, the output is an encrypted image with hidden data as illustrated in Fig. 2. ITE and ETI basically perform data hiding and encryption in a sequential manner. I2E aims to severely distort the quality of the image, which is going completely against the traditional principle of data hiding, *i.e.* to maintain the quality of the image [23]. On the other hand, E2I restricts the operations in the conventional perceptual encryption technique, where the ciphertext image is generated based on the data to be hidden. For example, the notion of a 'natural' state of some selected elements (*e.g.* group of pixels) is identified. The elements are then processed (*e.g.* cycle-shift) and the distance between the

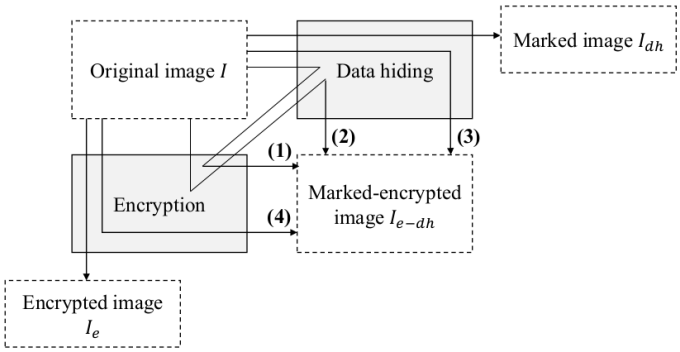

Fig. 2: Four typical ways to deploy data hiding and encryption in an image: 1) Insert-then-encrypt (ITE), 2) Encrypt-then-insert (ETI), 3) Insert-to-encrypt (I2E), and 4) Encrypt-to-insert (E2I) [22].

output state and the 'natural' state is exploited to hide data [24, 25]. For the purpose of this survey, we focus on ETI, because it is the most popular and natural way in combining data hiding and encryption. Unless specified otherwise, we use the terms *secret message* and *payload data* interchangeably. We also use the term *marked* image to indicate that the image contains embedded data.

The rest of this paper is organized as follows. Section 2 describes the different classes and characteristics of reversible data hiding in encrypted image (RDHEI) methods. A survey of the methods, including a classification, is then presented in Section 3. Section 4 reports the experimental results and comparisons with other state-of-the-art methods. Then, Section 5 gives some perspective works for the next 12 years. Finally, this paper is concluded in Section 6.

2. Different classes and characteristics

In recent years, many methods of RDHEI have been developed. Due to their diversity, several classes and characteristics can be defined. In Section 2.1, we start by defining the different properties that classify several state-of-the-art methods, namely: the trade-off between the payload and the quality of the reconstructed image, as well as the approaches that can be used for the encoding and decoding phases. We then describe the classical approaches for image encryption in RDHEI methods in Section 2.2. Indeed, while stream encryption is generally performed to protect the confidentiality of the original data in RDHEI methods, some RDHEI methods use block encryption with, for example, the Advanced Encryption Standard (AES) [13] algorithm, or employ a public-key cryptosystem, such as that defined by Paillier [26]. Finally, in Section 2.3, we detail the criteria used to evaluate their performance. In fact, a RDHEI method can be evaluated according to the number of embedded bits in terms of payload, the bit error rate (BER) during message extraction, and the visual quality after reconstruction in comparison to the original image. In addition, we also focus on assessing the visual security level of the marked encrypted image through in-depth statistical analysis.

2.1. Properties

The properties for defining and categorizing RDHEI methods are detailed in this section. The notion of trade-off between the payload and the quality of the reconstructed image are developed in Section 2.1.1. Then, the approaches for the encoding and decoding phases are described in Section 2.1.2 and Section 2.1.3, respectively.

2.1.1. Quality vs capacity: a real trade-off

In RDHEI methods, there is a trade-off between the number of secret bits embedded in the encrypted image and the quality of the reconstructed image after decryption of the marked encrypted image. Thus, two types of methods can be defined based on the

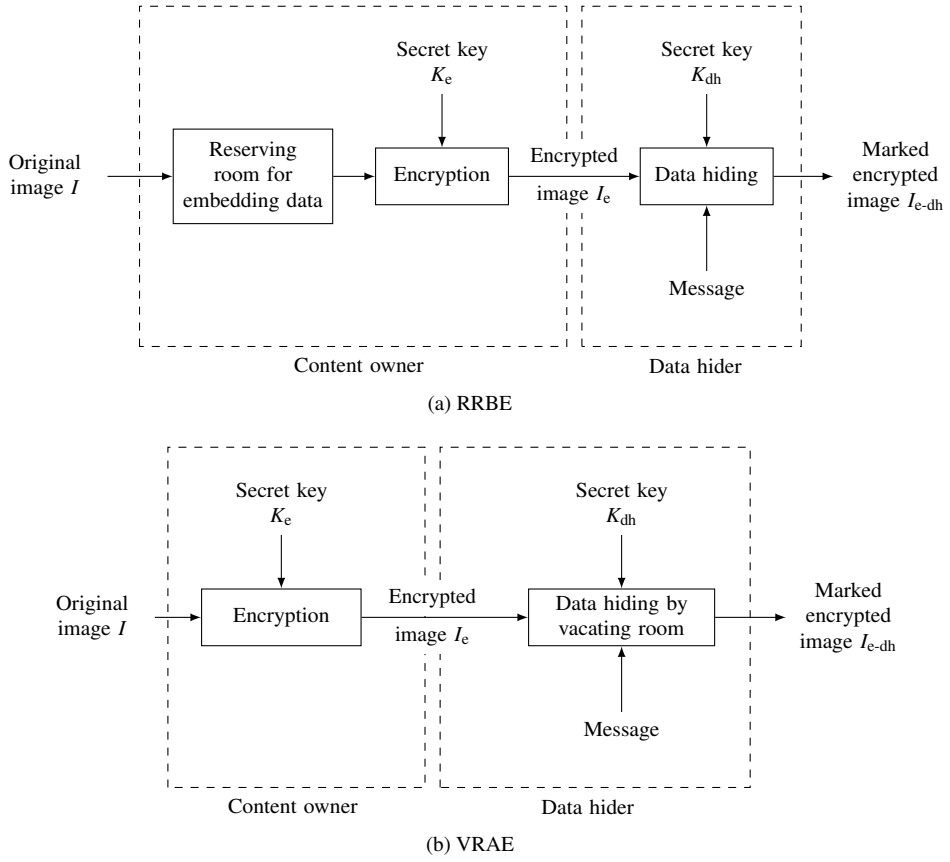

Fig. 3: Two possible encoding schemes: a) Reserving room before encryption (RRBE), b) Vacating room after encryption (VRAE).

number of bits (*i.e.* length of a secret message) that can be embedded. Specifically, low-capacity RDHEI methods are those with a payload of less than 0.5 bit-per-pixel (*bpp*). Conversely, a method is said to have a "high-capacity" when the payload is ≥ 1 *bpp*. We note that, until 2018, none of the state-of-the-art RDHEI methods could embed more than 1 *bpp*. In addition, after decryption of the marked encrypted image, some methods allow the secret message to remain in the decrypted (*i.e.* plaintext) image. In this case, the reconstructed image must be similar to the original plaintext image. In addition, a method is "fully reversible" when the original image can be reconstructed without any loss after the extraction of the secret message. Finally, it should be noted that longer secret messages imply a higher risk of quality degradation for the image reconstructed during the decoding phase, and vice versa.

2.1.2. Two approaches for the encoder

RDHEI methods can be divided into two categories based on the encoding process, namely approaches based on freeing up space for message embedding before encryption (RRBE, Reserving Room Before Encryption) or freeing space for message embedding after encryption (VRAE, Vacating Room After Encryption), as illustrated in Fig. 3. For the former, the original image is pre-processed by its owner before encryption, in order to release some space to hide a secret message. The image is then encrypted and another person, for example, the manager of a cloud server can embed parts of the secret message at the specific positions dedicated to this purpose [27, 28, 29, 30]. For the latter, the content of the original image is encrypted directly by its owner without any pre-processing. The manager of the cloud server then modifies the encrypted data in order to hide the secret message [15, 31, 32, 33, 34]. Both approaches are effective, but each has certain limitations. Specifically, RRBE-based methods are able to embed longer secret messages in general, but a pre-processing phase is required prior to encryption. This can be a problem and is impractical if the image owner does not foresee that the encrypted image needs to be analyzed or processed later. Instead, for VRAE-based methods, the

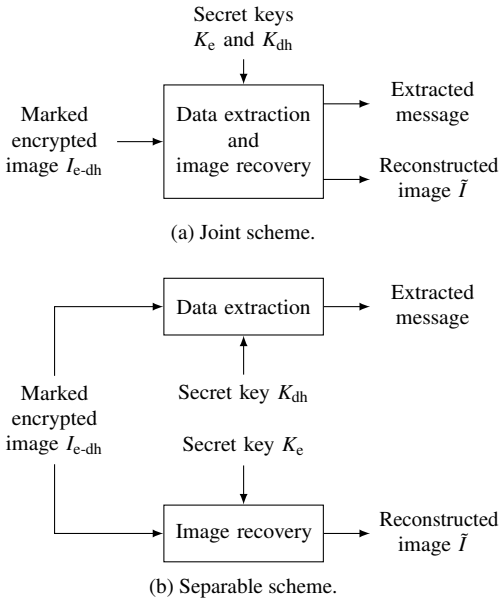

Fig. 4: Two possible decoding schemes for the extraction of the secret message and the reconstruction of the original image: a) Joint or b) Separable.

recipient of the marked encrypted image must predict the content of the original image in order to reconstruct it. Therefore, the reconstructed image is usually an estimation of the original image and perfect reconstruction cannot be achieved. In addition, in order to minimize the introduced distortion, a large number of secret bits cannot be embedded.

2.1.3. Two approaches for the decoder

During the decoding phase, the extraction of a message and the reconstruction of an original image can be carried out jointly or separately, as shown in Fig. 4. If the decoding is *joint*, this means that the plaintext image cannot be obtained without knowing the data hiding key. Indeed, by using only the encryption key, only a degraded version of the original image can be obtained [15]. Furthermore, in some methods, knowing only the data hiding key does not allow us to extract the secret message [32]. However, if the decoding is *separable*, the operations required for extracting the message and reconstructing the original image can be performed independently. Here are the resulting two scenarios:

- A clear image with the embedded secret message is obtained. This image is very similar to the original image, but not identical [33];
- The original image can be perfectly reconstructed, which does not contain the secret message [30].

When a user only knows the data hiding key, they can extract the secret message directly from the encrypted image. Note that the secret message can also be extracted in the clear domain, if the method is designed to allow the secret message to be kept in the clear domain.

2.2. Classical encryption approaches

The most commonly used technique, which is stream encryption, is described in Section 2.2.1. Symmetric block encryption algorithms are then presented in Section 2.2.2. Finally, homomorphic encryption algorithms are detailed in Section 2.2.3.

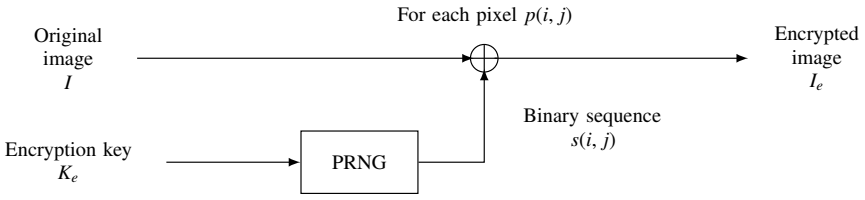

Fig. 5: Stream encryption.

2.2.1. *Stream encryption*

In most recent state-of-the-art methods [32], the original image I of size $m \times n$ pixels $p(i, j)$ $(0 \leq i < m, 0 \leq j < n)$ is encrypted by using a stream encryption algorithm. As shown in Fig. 5, an encryption key K_e is used as a seed to a pseudo random number generator (PRNG). This PRNG provides a sequence of pseudo-random numbers $s(i, j)$ in the form of bytes. Thus, the encrypted pixels $p_e(i, j)$ are computed by performing bitwise exclusive-or (XOR, denoted by the symbol $\oplus$) operation using the binary sequence $s(i, j)$ and the pixel value $p(i, j)$:

$$p_e(i, j) = s(i, j) \oplus p(i, j). \tag{1}$$

Since the encryption operation is completely reversible and does not cause any under/overflow, it is then possible to recover the clear image I from the encrypted image I_e without any alteration. We note that the pseudo-random sequence can be generated by using a cryptographically secure pseudo random number generator (CSPRNG), a chaotic generator [35], or by using the AES algorithm [13] in output feedback (OFB) mode for example.

2.2.2. *Symmetric block encryption*

Some other state-of-the-art methods exploit the correlation within the pixel blocks of an image in the clear domain [15]. As a result, they use symmetric block encryption algorithms to preserve the structure of pixel blocks in the encrypted image. The most commonly used symmetric block encryption algorithm is AES [13]. It was designed in 1999 by Joan Daemen and Vincent Rijmen, and it consists of a set of operations, which are repeated over several iterations called *rounds*. The number of rounds depends on the size of the encryption key, for example, 10 repetition cycles for 128-bit keys, 12 repetition cycles for 192-bit keys, and 14 repetition cycles for 256-bit keys. In order to encrypt a 128-bit sequence, the AddRoundKey operation is first applied. Each byte in the sequence is combined with an associated block in the round key using the XOR operation. Then, during each round, four different steps are performed, namely, SubBytes, ShiftRows, MixColumns and AddRoundKey. The SubBytes operation is a non-linear substitution step, where each byte is substituted by another according to a conventionally defined table. The ShiftRows operation is a transposition step where the last three rows of the block are cyclically shifted. The MixColumns is a linear mixing operation that operates on the columns of the block, combining four bytes of each column. Finally, the last round consists of the same operations, but without performing the MixColumns. Furthermore, the AES algorithm can support different encryption modes, such as Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) or Counter (CTR).

2.2.3. *Homomorphic encryption*

Many RDHEI methods exploit the probabilistic and homomorphic properties of public key cryptosystems. A function $f(\cdot)$ is said to be homomorphic if the following condition is verified:

$$f(x_1 \vartriangle x_2) = f(x_1) \square f(x_2), \tag{2}$$

where $\triangle$ and $\square$ are some arithmetic operations.

By extension, an encryption algorithm $\mathcal{E}(\cdot)$ (with associated decryption algorithm $\mathcal{D}(\cdot)$) is said to be homomorphic if the encrypted versions of two plaintext messages m_1 and m_2 are known, then it is possible to obtain the encrypted version of the output for an operation between these two messages:

$$\mathcal{D}(\mathcal{E}(m_1 \triangle m_2)) = \mathcal{D}(\mathcal{E}(m_1)) \square \mathcal{D}(\mathcal{E}(m_2)). \tag{3}$$

Note that $\triangle$ and $\square$ can denote addition, subtraction or multiplication. They are not necessarily the same between plaintext messages and their encrypted versions.

Asymmetric cryptosystems are often homomorphic:

- River-Shamir-Adleman (RSA) [36] and El Gamal's cryptosystem are partially homomorphic to multiplication: $\mathcal{D}(\mathcal{E}(m_1) \cdot \mathcal{E}(m_2)) = m_1 \cdot m_2$, where $\mathcal{D}(\cdot)$ is the decryption algorithm associated to $\mathcal{E}(\cdot)$.
- Paillier's cryptosystem [26] is an additive homomorphism: $\mathcal{D}(\mathcal{E}(m_1) \cdot \mathcal{E}(m_2)) = m_1 + m_2$.

The main advantage of the homomorphism property is that it allows operations to be performed in the encrypted domain without revealing any information about the content in the clear domain. Most RDHEI methods adopt Paillier's cryptosystem [26] to encrypt the image. Specifically, to generate the public and private keys, two prime numbers p and q are chosen such that:

$$\gcd(pq,\ (p-1)(q-1)) = 1, \tag{4}$$

where gcd refers to greatest common divisor. The values $\eta = pq$ and $\lambda = \text{lcm}((p-1),(q-1))$ are then computed, where lcm refers to the least common multiple. Next, an integer $g \in (\mathbb{Z}/\eta^2\mathbb{Z})^*$ is selected such as:

$$\exists\, \mu \mid \mu\ = (L(g^{\lambda} \bmod (\eta^2)))^{-1} \bmod (\eta), \tag{5}$$

where $L(\cdot)$ is defined as:

$$L(x) = \frac{x-1}{\eta},\ \text{where } x \in \mathbb{N}^*. \tag{6}$$

The public key is (η, g) and the private key is (λ, μ). Then, we consider a to-be-encrypted message ω such that $0 \leq \omega < \eta$. In the case of image encryption, ω usually corresponds to a pixel or a pixel block. For the message encryption, a random number r is generated, with $r \in (\mathbb{Z}/\eta\mathbb{Z})^*$. Note that choosing such r guarantees the non-determinism property of Paillier's cryptosystem. Finally, the encrypted message c is calculated:

$$c = \mathcal{E}(\omega) = g^{\omega} \times r^{\eta} \bmod (\eta^2), \tag{7}$$

where $\mathcal{E}(\cdot)$ is the encryption function of Paillier's cryptosystem. We note that the squaring of η leads to size expansion of two in the encrypted message when compared to its original counterpart.

From the encrypted message c, the original message in clear ω can be reconstructed:

$$\omega = \mathcal{D}(c) = L(c^{\lambda} \bmod (\eta^2)) \times \mu \bmod (\eta), \tag{8}$$

where $\mathcal{D}(\cdot)$ is the encryption function of Paillier's cryptosystem.

2.3. Evaluation criteria

In this section, we first discuss the differences between embedding capacity and payload (Section 2.3.1). Then, we describe the notion of bit error rate (Section 2.3.2). We also present the metrics commonly adopted for visual quality evaluation (Section 2.3.3). Finally, the metrics used for visual security level are developed in Section 2.3.4.

2.3.1. Amount of embedded bits

The amount of embedded bits is expressed in bits-per-pixel (bpp). For an image whose pixels are encoded using 256 grey-levels, this quantity will be between 0 bpp and 8 bpp. Furthermore, we make a distinction between embedding capacity and payload. Specifically, the embedding capacity refers to the total number of bits that can be embedded in an image using the RDHEI method. Payload refers to the total number of bits of the message that can be embedded into an image. Therefore, the payload can be significantly less than the embedding capacity, especially when a RDHEI method needs to sacrifice some storage to encode additional information, such as information needed to handle overflow [37], or to flag pixels that cannot be predicted correctly (*i.e.* large prediction errors) [30, 38].

2.3.2. Bit error rate (BER)

The BER when extracting the message from the marked encrypted image (or the marked decrypted image when it can be obtained) is calculated by dividing the number of badly reconstructed bits by the total number of bits of the secret message. So, the BER must be as small as possible to ensure proper transmission of the secret data embedded in the encrypted image. In most recent state-of-the-art methods, the BER is zero, which means that the secret message is extracted without any errors.

2.3.3. Visual quality

The visual quality between the reconstructed image (*i.e.* decrypting the marked encrypted image) and its original counterpart can be assessed by using two commonly deployed metrics in image coding and compression, namely PSNR and SSIM. Note that the quality of the reconstructed image can be evaluated before and after the secret message extraction. Indeed, if the RDHEI method allows the message to remain in the decrypted image, it is interesting to evaluate the distortion introduced by embedding the message in the original image directly. Furthermore, it should be noted that some methods are not reversible and it is impossible to recover the original image without some alteration after extracting the secret message.

Peak-Signal-to-Noise Ratio (PSNR): PSNR is commonly used to measure the quality of the reconstructed image against the original image, which is computed as:

$$\mathrm{PSNR} = 10 \cdot \log_{10} \frac{(2^l - 1)^2}{\frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (p(i,j) - p'(i,j))^2}, \tag{9}$$

where $p(i, j)$ is a pixel from the original image and $p'(i, j)$ is the corresponding pixel in the reconstructed image, both of the same size and whose pixels are encoded by using 2^l grey-level values. PSNR is measured in the unit of decibels (dB).

In general, images are encoded by using 256 grey-level values ($l = 8$) and hence $2^l - 1 = 255$. In this case, between two very different images, PSNR is about 10 dB, and between two similar but noise-altered images, PSNR is about 15 dB. On the other hand, when two images are highly similar, PSNR is more than 30 dB. Finally, in the case of a perfect reconstruction, *i.e.* when the two images are exactly the same, its value tends toward $+\infty$. Note that many RDHEI methods are described as reversible although the value of PSNR is about 50 dB. The major drawback of PSNR is that it does not take into account the quality perception of the human

visual system (HVS). Thus, it cannot be considered as a totally objective measure.

Structural Similarity (SSIM) [39]: SSIM allows the assessment of structural similarity between two images, rather than relying solely on pixel-to-pixel differences as in PSNR. The underlying assumption in SSIM is that the HVS is more sensitive to changes in image structure. In particular, it is modeled to detect changes in terms of loss of correlation, luminance distortion and contrast distortion between two images. Specifically, SSIM is computed as follows:

$$\mathrm{SSIM}(x,y) = \frac{(2E(x)E(y)+\gamma_1)\,(2Cov(x,y)+\gamma_2)}{(E(x)^2+E(y)^2+\gamma_1)\,(V(x)^2+V(y)^2+\gamma_2)}, \tag{10}$$

where x and y are the windows of the two images, $E(x)$ is the mean of the x set, $V(x)$ the variance of the x set, $Cov(x,y)$ the covariance between the x and y sets, $\gamma_1 = (0,01 \times (2^l-1))^2$, and $\gamma_2 = (0,03 \times (2^l-1))^2$.

This formula is applied to the co-located windows (*i.e.* a block of pixels with the same coordinates) in both images, and the average of the values obtained for these co-located windows yields the final SSIM score. This score ranges between -1 and 1. It is equal to 1 when the two images are exactly the same. The value -1 is reached when one image is the complement (*i.e.* $y = 255 - x$) to the other. However, for data hiding, the common range of observation ranges between 0 and 1. Note that it is often necessary to observe at least three decimal values for a significant SSIM measurement.

2.3.4. Visual security level

Until now, no specific metrics have been defined to evaluate the visual security level of marked encrypted images. Therefore, based on the work of Preishuber *et al.* [40], we describe the metrics used in many papers to experimentally demonstrate the visual security of image encryption methods. We conclude that a good level of visual security is achieved in the marked encrypted image when the message embedding phase does not impact the security level of the encryption method. Nevertheless, if the embedded message is itself encrypted, embedding this message into an encrypted image is similar to embedding "noise into noise". So, the encrypted message cannot *a priori* be detected in the encrypted domain.

Correlation coefficient: A common metric is to observe the correlation between pixels in the horizontal, vertical and diagonal directions. M pairs of neighboring pixels (x_i, y_i) in all three directions, with $x_i \in x$ and $y_i \in y$, are thus chosen for the calculation of the correlation coefficient:

$$corr_{x,\,y} = \frac{\frac{1}{M}\sum_{i=1}^{M}(x_i - E(x)) \times (y_i - E(y))}{\sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_i - E(x))^2}\sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - E(y))^2}}. \tag{11}$$

where $E(x)$ is the average of the set x.

The value of this correlation coefficient ranges from -1 to 1, where -1 and 1 indicate a high correlation while 0 indicates a lack of correlation. Since the neighboring pixel values in the clear domain are highly correlated, $corr_{x,\,y}$ is generally high in the original image. However, it should be close to zero in the encrypted domain.

Shannon entropy [41]: Shannon entropy is a measure of the amount of information commonly used to evaluate the randomness of the pixel distribution of a marked encrypted image. It is computed as follows:

$$H(I) = -\sum_{k=0}^{N-1} P(\alpha_k)\log_2(P(\alpha_k)), \tag{12}$$

where I is an image of $m \times n$ pixels encoded by using N values α_k ($0 \leq k < N$) and $P(\alpha_k)$ is the probability associated with α_k.

The entropy value is expressed in bpp and it ranges from 0 bpp to $\log_2(N)$ bpp, where the value of $\log_2(N)$ is achieved only when the pixel distribution is perfectly uniform. In general, images are encoded by using 256 grey-level values and hence the maximum entropy is then $\log_2(256) = 8$ bpp. The entropy value of a marked encrypted image should be very close to the maximum entropy value, *i.e.*, $\log_2(N)$.

χ^2-test: The uniformity of the pixel distribution of a marked encrypted image can also be assessed by using the chi-square (χ^2-) test, which is computed as:

$$\chi^2 = N \sum_{k=0}^{N-1} \left(P(\alpha_k) - \frac{1}{N} \right)^2, \tag{13}$$

where the pixels of the image are coded by using N values α_k ($0 \leq k < N$) and $P(\alpha_k)$ is the probability associated with α_k.

The lower the value obtained, the closer the pixel distribution of the marked encrypted image is to the uniform distribution, thus indicating a higher level of visual security. Note that the square root of the χ^2 value is often considered.

Number of Changing Pixel Rate (NPCR) [42]: NPCR between two images of size $m \times n$ pixels $p(i, j)$ and $p'(i, j)$ ($0 \leq i < m$, $0 \leq j < n$) is given by:

$$\text{NPCR} = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} d(i, j)}{m \times n} \times 100, \tag{14}$$

where $d(i, j)$ is defined as:

$$d(i, j) = \begin{cases} 1, \text{ if } p(i, j) \neq p'(i, j), \\ 0, \text{ otherwise.} \end{cases} \tag{15}$$

It is expressed in % and is used to indicate how much a marked encrypted (or simply marked) image differs from the original image. Therefore, the closer its value is to 100%, the more mismatches between the two images and so the higher the level of visual security.

Unified Averaged Changed Intensity (UACI) [42]: UACI is also used to measure the differences between two images of $m \times n$ pixels, where the pixels $p(i, j)$ and $p'(i, j)$ ($0 \leq i < m,\ 0 \leq j < n$) are encoded by using 256 grey-level values. It is computed as follows:

$$\text{UACI} = \frac{100}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{|p(i, j) - p'(i, j)|}{255}. \tag{16}$$

Similar to NPCR, UACI is expressed in %, where the higher its value, the higher the level of visual security. Note that its ideal value depends on the tonal range of the image. This metric can also be used to test the sensitivity of the key used during encryption. Specifically, the original image is encrypted by using two keys that differ by only one bit. The corresponding encrypted images are then compared. In this case, the optimal value of UACI is 33.33% [40]. However, if UACI is used to compare an original image in clear and its marked encrypted version, the value is often lower. Note that there is no statistical decision criterion for this test, and the observation of optimal values is purely experimental.

3. Survey of the methods

This section aims to classify the existing RDHEI techniques based on the underlying image processing mechanisms involved. Specifically, the classes include: image partition-based (Section 3.1), histogram shifting or pixel value ordering (PVO)-based (Section 3.2), re-encoding-based (Section 3.3), prediction-based (Section 3.4) and public-key cryptography-based (Section 3.5).

3.1. Image partition-based methods

In many RDHEI methods, the pixels in an image are **divided into two groups**. Specifically, the pixels in the first group are reserved to embed a secret message, while those in the second group are not marked, but used to reconstruct the original image. Consequently, this **image partitioning process is performed before or after encryption**, depending on the method in use. Some authors perform partitioning in a pseudo-random way by using the embedding key [32]. Instead, other techniques analyze the properties of the pixels in the clear image [27].

3.1.1. Exploitation of the data hiding key [32]

In one of the earliest RDHEI methods, Zhang *et al.* proposed to encrypt an original image by using a stream encryption algorithm [32]. The encrypted image is further divided into non-overlapping blocks of $s \times s$ pixels, where one bit of the secret message is hidden into each block. Within each block, the pixels are then partitioned into two groups, namely, S_0 and S_1 using an embedding key. If the secret message bit to be embedded is '0' (resp. '1'), the three least significant bits (LSBs) of each pixel in S_0 (resp. S_1) are flipped. During the decoding phase, the marked encrypted image is decrypted in order to obtain an approximation of the original image. The five most significant bit planes (MSBs) are perfectly reconstructed and only the 3 LSBs are altered. Then, a function is used to evaluate the inconsistencies in the reconstructed pixels within the S_0 and S_1 groups. In this way, the secret message can be extracted, one bit per block, and the original configuration of each pixel block can be retrieved. Increasing the size of the blocks increases the chances of perfectly reconstructing the original image. However, Zhang's method is not reversible because it does not guarantee a perfect reconstruction of the original image. Furthermore, the payload is low (less than 0.1 bpp) since only one bit is embedded for each $s \times s$ block. Improvements to this method have then been made in the following literature [31, 33].

3.1.2. Exploitation of a fluctuation function [27]

Ma *et al.* are the first to propose an RRBE method, taking the opposite line to all of the other state-of-the-art methods developed, up to this point in time [27]. The authors start by splitting the original image into blocks. Within each block, the correlation between the pixels is evaluated by using a fluctuation function. The blocks are further partitioned into two groups, namely, A and B. Specifically, group A is composed of textured blocks while group B is composed of relatively homogeneous blocks. The blocks in group A are placed at the beginning of the image and those in B are placed afterwards. In order to release space for the secret message embedding, the LSB plane of A is embedded into pixels in group B in the clear by using histogram shifting. The resulting image is then encrypted by using a stream encryption algorithm and the number of pixels that can be marked is stored in the LSBs of the first A pixels. With this information, the secret message can be embedded simply by substituting the LSBs of the remaining pixels in A. We note that the first three LSBs of each pixel can be used. The total payload of the marked encrypted image can reach 0.5 bpp. Therefore, in comparison to previous state-of-the-art methods, the payload is ten times greater. Finally, during the reconstruction phase, the secret message extraction and the original image reconstruction operations can be performed separately. Based on this idea, another method has been proposed by Zhang *et al.* [43].

3.2. Histogram / PVO-based methods

Many RDHEI methods based on **histogram shifting** have been developed due to their simplicity in implementation and their ability to produce **very high quality images after decrypting the marked encrypted image**. In natural images in the clear domain, **neighboring pixels are highly correlated**. As a result, the distribution of differences between neighboring pixels is modelled by a zero-centered Laplacian distribution. In fact, most histogram-based methods [44, 45, 46] rely on this particular property. Consequently, **these statistical characteristics can be exploited during the secret message embedding phase** in histogram shifting-based RDHEI methods [37, 45, 47].

3.2.1. Pixel difference or prediction error histogram [37]

In 2016, Huang *et al.* discovered that the previously proposed algorithms for embedding secret data in the clear domain cannot be applied in the encrypted domain [37]. Indeed, classical encryption methods do not allow the correlation between neighboring pixels to be maintained without introducing security flaws. In their work, the authors introduce a new strategy to encrypt an image with the objective being that the conventional data hiding algorithms designed in the clear domain can be deployed directly to embed data in the encrypted image. Specifically, the original image is split into non-overlapping blocks. Within each block, all pixels are encrypted by applying the XOR operation using the same pseudo-randomly generated byte. The encrypted blocks are then pseudo-randomly permuted. Note that pixels within the same block are not scrambled, but only the order of the blocks are changed. With this encryption method, the statistical properties of the clear image, especially the histogram of pixel differences or prediction errors, are preserved. Therefore, the conventional data hiding algorithms in the clear domain can be applied in the encrypted domain, but the embedding capacity is limited by the handling of under/overflow problems.

3.2.2. Homomorphism and PVO [45]

Xiao *et al.* proposed to adapt the concept of PVO, defined in the clear domain by Li *et al.* [48], to embed data in the homomorphic encrypted domain [45]. The authors start by arguing that by adapting the homomorphic encryption method, it is possible to obtain a histogram of the differences between pixels. The histogram follows a zero-centered Laplacian model in the encrypted domain, which is identical to the histogram obtained in the clear domain. Specifically, the encrypted image is split into 2×2 non-overlapping blocks. The pixels in each block (4 in total) are collected and re-arranged in ascending order, which are denoted as c_1, c_2, c_3, c_4. Then, the pixel with the lowest (c_1) and highest (c_4) value are selected to embed data. Specifically, the difference $\Delta = c_4 - c_3$ is calculated, and data hiding is performed depending on d:

$$c'_z = \begin{cases} c_z + w & \text{if } \Delta = 1 \\ c_z + 1 & \text{if } \Delta > 1 \\ c_z + w & \text{if } \Delta = 0 \\ c_z + 1 & \text{if } \Delta < 0, \end{cases} \tag{17}$$

where z is the sorted pixel index, c'_z is the embedded-encrypted pixel, and $w \in \{0, 1\}$ is the secret message. Similar operations are applied for the lowest value, namely c_1.

In this method, the pixel order remains unchanged after embedding data. Looking from the perspective of histogram, the difference bins of value '0' and '1' are expanded for data hiding while other bins are shifted. This method solves the inseparable and data expansion problems of other similar methods in the encrypted domain. Nevertheless, as reported in [45], this method can only hide up to 0.2 bpp. In fact, one of the biggest drawbacks of histogram-based methods is the over/underflow problem after histogram modification. To mitigate this problem, the authors in [45] employed a location map to record unusable blocks, which are skipped during data embedding. Hence, the introduction of a location map significantly impacts the effective payload of the proposed method.

3.2.3. Pixel value histogram [47]

In 2019, Ge *et al.* proposed a block-based histogram shifting method in encrypted domain [47] to enhance the embedding capacity of Xiao *et al.*'s method [45]. Instead of using the histogram of pixel differences, they directly modify the pixel value by using two reference pixels within the block. Unlike Xiao *et al.*'s method [45], their approach does not restrict the assignment of the reference pixels to be the lowest and highest values. Instead, the reference pixels are pseudo-randomly chosen based on the user's key. Let c_l and c_h denote the two selected reference pixels in an encrypted block such that $c_l < c_h$. Then, for each of the non-reference pixels in the same block (denoted by $\hat{c}_y$), data hiding is performed by using the following equation:

$$\hat{c}'_y = \begin{cases} \hat{c}_y - 1 & \text{if } \hat{c}_y < c_l \\ \hat{c}_y - w & \text{if } \hat{c}_y = c_l \\ \hat{c}_y & \text{if } c_l < \hat{c}_y < c_h \\ \hat{c}_y + w & \text{if } \hat{c}_y = c_h \\ \hat{c}_y + 1 & \text{if } \hat{c}_y < c_h, \end{cases} \tag{18}$$

where $\hat{c}'_y$ is the embedded-encrypted pixel with index y.

The same data hiding operation is performed on all the blocks in the entire image. In addition, multiple rounds of embedding can be performed on an encrypted image to further increase the embedding capacity.

For the comparison of single-level embedding, Ge *et al.* can only embed half the payload achieved by Xiao *et al.*'s method [45], despite the implementation of a pixel-shifting strategy. Indeed, the size of the location map is more complex than that used in [45], because each pixel which causes overflow/underflow needs to be recorded. Nevertheless, in the multi-level embedding mode, it is possible to embed up to 0.8 *bpp* while sacrificing some image quality in the reconstructed image in the clear domain. Note that Long *et al.* further improved these results in [49].

3.3. Re-encoding-based methods

As explained, in some state-of-the-art methods, **re-encoding** can be applied to the image data, **before or after encryption**, in order to optimize the number of bits needed for its representation. Thanks to this re-encoding phase, some **gain in memory space** is achieved. Thus, **the released space is used to embed the secret message**. A few algorithms implement re-encoding. In particular, Mustafa *et al.*'s method which is based on Golumb-Rice codewords [50], Qian *et al. et al.*'s method which is based on distributed source coding [51], and Cao *et al.*'s method which is based on sparse coding [28], have shown interesting performances.

3.3.1. Golomb-Rice codewords [50]

Mustafa *et al.* proposed a novel data hiding method in which any encrypted (in fact, any) signal encoded in binary representation can be further processed to embed data [50]. This is an example of VRAE method. Specifically, given the bitstream of any content regardless if it is in plaintext and encrypted form, the bitstream is uniformly divided into segments of equal length *len*, and each segment (*i.e.* imaginary codeword) is mapped to a unique Golomb-Rice codeword (*grc*). Each *grc* consists of two parts, namely, the quotient part *quo* and the remainder part *rem*, which are separated by a delimiter δ. Each *grc* takes the general form of:

$$grc = quo \mid \delta \mid rem, \tag{19}$$

where '|' is the concatenation operator, *quo* is in unary representation, δ is a fixed character (*e.g.* '1'), and *rem* is coded raw in binary representation.

The first 6 *grc*'s for a divisor $rem = 2^2 = 4$ are: 1$\underline{00}$, 1$\underline{01}$, 1$\underline{10}$, 1$\underline{11}$, 01$\underline{00}$, and 01$\underline{01}$, where the remainder is underlined.

The regular-pattern of *grc* (*i.e.* the quotient and delimiter parts) is exploited to embed data. Specifically, the quotient part of each codeword, which is the only part that has variable length, causes bit stream size increment. To maintain the original bit stream size, the quotient part of each codeword is extracted and treated as part of the secret message. The augmented secret data is then embedded by forming new codewords of constant length *len*. For each original codeword, each segment of the augmented data (with length $len-\text{length}(rem)$) becomes the new quotient quo' and delimiter δ' parts of the codeword, while the remainder part remains unchanged, essentially $grc' = quo' \mid \delta' \mid rem$. While Mustafa *et al.*'s method is applicable to any signal and completely reversible, it is neither separable nor commutative.

3.3.2. Distributed coding source [51]

In 2016, Qian *et al.* proposed to use distributed source coding in a RDHEI method [51]. During the encoding phase, the original image is first encrypted by using a stream encryption algorithm. The encrypted image I_e of pixels $p_e(i,j)$, with $0 \leq i < m$ and $0 \leq j < n$, is then divided into four sub-images $I_e^{(k)}$ $(1 \leq k \leq 4)$, whose pixels $p_e^{(k)}(i,j)$, with $0 \leq i < \frac{m}{2}$ et $0 \leq j < \frac{n}{2}$, are such that:

$$\begin{cases} p_e^{(1)}(i,j) = p_e(2i-1, 2j-1), \\ p_e^{(2)}(i,j) = p_e(2i-1, 2j), \\ p_e^{(3)}(i,j) = p_e(2i, 2j-1), \\ p_e^{(4)}(i,j) = p_e(2i, 2j). \end{cases} \tag{20}$$

Note that the decryption of each of the $I_e^{(k)}$ sub-images results in a thumbnail of the original image. After obtaining the sub-images $I_e^{(k)}$, three MSB planes of $I_e^{(2)}$, $I_e^{(3)}$, and $I_e^{(4)}$ are first scrambled and then compressed with low-density parity-check codes (LDPC) [52]. This compression has the effect of generating some space for data hiding. Here, the decoding phase is completely separable. The sub-image $I_e^{(1)}$, which has not been modified, is decoded and up-sampled by using bilinear interpolation in order to obtain a reference image to reconstruct the marked image in the clear domain. Furthermore, to reconstruct the original image, the LDPC codes are decoded by a sum-product algorithm [53].

3.3.3. Sparse coding [28]

In order to accommodate a large payload, Cao *et al.* suggest utilizing sparse coding in their RDHEI method [28]. The RRBE encoding phase consists of three stages. First, the original image is divided into patches, which are then represented by using a redundant dictionary, *i.e.* using sparse encoding. Next, the patches with the smallest residual errors (the homogeneous ones) are selected for data hiding, where they are represented by the sparse coefficients. At the same time, residual errors are encoded and concealed in the patches which are not selected for embedding by using a classical data hiding algorithm for images in clear. Finally, stream encryption is performed to protect the data in the clear domain. Once the image is encrypted, the secret message is divided into segments and embedded into the previously released space. Ultimately, the decoding phase is separable and reversible. The original image can also be reconstructed without any loss by using the residual errors extracted from the unmarked patches, and the secret message can be retrieved.

3.4. Prediction-based methods

In some state-of-the-art methods, during the encoding phase, selected bits of certain pixels of the encrypted image are substituted by segments of the secret message. Consequently, their original value is lost and **has to be predicted** during the decoding phase in order to achieve **high quality reconstruction** of the original image in the clear domain. This **prediction** can be achieved by exploiting the **differences between a block of pixels in the clear domain and its encrypted version** [15], or the **strong correlation between a pixel and its neighborhood in the clear domain** [54, 30].

3.4.1. Prediction based on statistical analysis [15]

In 2008, Puech *et al.* proposed one of the first RDHEI methods [15]. During the encoding phase, the method opts for a VRAE approach. The original image is encrypted in blocks of 16 grey-level pixels (128 bits) using the AES algorithm in ECB mode. One bit of the secret message is then embedded into each block of the encrypted image, which translates to an embedding capacity of 0.0625 bpp. Note that a data hiding key is used as the seed for a PRNG to determine the to-be-marked pixels and the location of the bit which is substituted by a bit of the message. Accordingly, the marked encrypted image is obtained when all the blocks have been processed. During the decoding phase, the extraction of the message is carried out by reading, using the data hiding key, the bits of the pixels that have been marked. However, after extraction, the pixels are still marked with the bits of the message, making it difficult to decrypt the image. To overcome this problem, a local analysis of the standard deviation in each block is performed. For each block of the marked encrypted image, by using the data hiding key to seed the PRNG, the marked bit is located and substituted by the two possible values of the original bit (*viz.* 0 and 1). Two configurations are then obtained and decrypted, namely, one corresponds to the associated block in the original plaintext image, while the other is erroneous and has the appearance of a fully encrypted block. The following hypothesis is then formulated: the standard deviation in an encrypted (incorrectly decrypted) block is greater than that of a clear (correctly decrypted) block. Consequently, the standard deviation associated with the two decrypted configurations is calculated. The configuration with the lower standard deviation value is considered to be the expected block in clear. Note that since the reconstruction of the original image requires the knowledge of the data hiding key, the decoding step is joint.

3.4.2. Prediction using interpolation [54]

Wu and Sun have developed a RDHEI method that can operate in two different ways, namely, joint approach, and separative approach [54]. In both approaches, the original image is first encrypted by using a stream encryption algorithm. Depending on the data hiding key, a subset of pixels is selected to perform data hiding. We note that the neighboring pixels of the selected pixels are used for prediction during the decoding phase. In the case of the joint approach, to embed one bit of the secret message, the LSBs of the selected pixels are flipped if the message bit is '1'. Otherwise they are kept unchanged if the message bit is '0'. During the decoding phase, the unmarked neighboring pixels are used to predict the original value of each marked pixel, as well as the value of the embedded bit. On the other hand, in the case of the separative approach, the LSBs of the selected pixels are substituted by one secret message bit. To reconstruct an approximation of the original image during the decoding phase, a median filter is then deployed. Subsequently, improvements have been made to both of these approaches [55, 56].

3.4.3. Prediction using MSB values [30]

In 2018, Puteaux and Puech proposed to use the MSB values instead of the LSB values to embed a secret message [30]. In fact, they stated that MSB substitution does not introduce artifacts in the encrypted domain and MSB prediction is easier than LSB prediction. Based on these two assumptions, they proposed two different high capacity reversible data hiding (HCRDH) approaches, namely, "corrected prediction errors" (CPE) and "embedded prediction errors" (EPE). In both approaches, all pixels in the clear image which cannot be correctly predicted by using their neighbors are first identified. Specifically, in the case of the CPE-HCRDH approach, the original image is pre-processed to avoid all prediction errors (PE). The pre-processed image is then encrypted and the data hider can blindly substitute all MSB values of the encrypted image by the secret message, *i.e.* 1 secret message bit per encrypted pixel. In this case, payload is 1 bpp and the reconstructed image corresponds to the pre-processed image, which is very close to the original one (PSNR $> 50\ dB$). However, in the case of the EPE-HCRDH approach, the original image is encrypted without any modification. After encryption, information about the location of all pixels which cannot be correctly predicted is embedded by MSB

substitution by the content owner. Then, the data hider can detect all bits which can be marked and replaces them by bits of the secret message. In this case, the payload is slightly lower than 1 bpp but perfect reversibility is achieved.

3.5. Public-key encryption-based methods

RDHEI methods **exploiting the homomorphic properties of public-key cryptosystems** can be classified into two categories depending on the encryption approach in use. Consequently, a distinction can be made between methods whose encryption is based on the **Paillier cryptosystem** [57], and those based on **post-quantic encryption and exploiting the Learning With Errors (LWE) problem** [58].

3.5.1. Paillier cryptosystem-based methods

In 2014, Chen *et al.* proposed the first RDHEI method based on the use of Paillier's cryptosystem [57]. Each pixel of the original image is divided into two distinct parts: the seven MSBs (hence, forming an integer value), and its LSB. Each part is then encrypted independently and one bit of the message is embedded into each pair of neighboring pixels. During the decoding phase, the receiver can reconstruct the entire secret message and the original image in clear by comparing all the decrypted pixel pairs. The main drawback of this method is that it does not handle the under/overflow problems. Shiu *et al.* then proposed a solution to this drawback [59], where they apply the concept of difference expansion to the homomorphic encrypted domain. We note that both methods are based on an RRBE approach.

VRAE methods relying on the use of the Paillier cryptosystem have been proposed by Wu *et al.* [60] and Zhang *et al.* [29]. The authors used the self-blinding property of the Paillier cryptosystem and value expansion in two algorithms. The first one allows data extraction in the encrypted domain. Conversely, in the other one, data extraction is achieved in the clear domain (after decryption). Furthermore, with the same objectives as Wu *et al.* [29], Zhang *et al.* also developed two different approaches: a lossless approach, where an embedded secret message can be extracted directly in the encrypted domain, and a separable approach, where it can be extracted from the clear reconstructed image. Moreover, Li and Li [61] suggested exploiting the homomorphic addition property of the Paillier cryptosystem and to resort to histogram shifting to perform data hiding steps. Xiang and Luo also proposed a separable homomorphic encryption-based method for RDHEI, involving the mirroring ciphertext group strategy [62]. By using such an approach, there is no pixel over-saturation in the clear domain after decryption, but the computational cost is high. In 2019, Zheng *et al.* proposed a lossless data hiding method based on homomorphic cryptosystem which achieves a high embedding rate through efficient mapping and skillful utilization of the expanded pixel values [63]. This method can realize a high embedding rate and low computational complexity, but it needs auxiliary data for the decoding step. Recently, Jiang and Pang presented an improved implementation of the Paillier cryptosystem based on the use of the Chinese Remainder Theorem [64]. This new implementation significantly improves the rapidity of encryption and decryption operations. After encryption, the authors observe the parity of the encrypted pixel pair to embed the secret message.

3.5.2. Post-quantic encryption-based methods

The first RDHEI method based on post-quantic encryption was proposed in 2016 by Ke *et al.* [58]. The authors argued that using post-quantic encryption and exploiting the LWE problem allows one to achieve the following: high security level, simple and fast implementation, as well as controllable redundancy for data hiding. Then, they set the encryption parameters and described their approach to achieve multi-level RDHEI. The latter is based on recoding the redundancy in the encrypted domain by using homomorphic operations. The main disadvantage of this approach is that it is not completely separable. In [65], Xiong *et al.* dealt

Table 1: Comparison of representative state-of-the-art methods according to the approach used for the encoding phase RRBE or VRAE, the type of decoding (joint or separable), reversibility (in the strict sense of the term, *i.e.* PSNR$\rightarrow +\infty$), and of the payload (in bpp or in bpb for methods indicated by a *).

Year	Method	Encoding	Decoding	Reversibility	Payload
2008	Puech *et al.* [15]	VRAE	Joint	No	< 0.1 bpp
2011	Zhang [32]	VRAE	Joint	No	< 0.1 bpp
2012	Hong *et al.* [31]	VRAE	Joint	No	< 0.1 bpp
	Zhang [33]	VRAE	Separable	No	< 0.1 bpp
2013	Ma *et al.* [27]	RRBE	Separable	Yes	< 0.5 bpp
2014	Chen *et al.* [57]	RRBE	Joint	No	< 0.001 bpb^*
	Wu and Sun [54] (1)	RRBE	Joint	No	< 0.5 bpp
	Wu and Sun [54] (2)	RRBE	Separable	No	< 0.5 bpp
2015	Shiu *et al.* [59]	RRBE	Joint	Yes	< 0.001 bpb^*
	Mustafa *et al.* [50]	VRAE	Separable	Yes	< 0.2 bpb^*
2016	Cao *et al.* [28]	RRBE	Separable	Yes	< 1 bpp
	Huang *et al.* [37]	RRBE	Separable	Yes	< 0.1 bpp
	Wu *et al.* [60] (1)	VRAE	Separable	No	< 0.5 bpb^*
	Wu *et al.* [60] (2)	VRAE	Joint	Yes	< 0.01 bpb^*
	Qian et Zhang [51]	VRAE	Separable	No	< 0.5 bpp
	Zhang *et al.* [29] (1)	VRAE	Joint	Yes	< 0.001 bpb^*
	Zhang *et al.* [29] (2)	RRBE	Separable	No	< 0.001 bpb^*
	Ke *et al.* [58]	VRAE	Joint	Yes	< 0.5 bpb^*
2017	Xiao *et al.* [45]	RRBE	Separable	Yes	< 0.5 bpp
	Dragoi *et al.* [55] (1)	RRBE	Joint	Yes	< 0.1 bpp
	Dragoi *et al.* [55] (2)	RRBE	Separable	Yes	< 0.1 bpp
2018	Dragoi and Coltuc [56] (1)	RRBE	Joint	Yes	< 0.1 bpp
	Dragoi and Coltuc [56] (2)	RRBE	Separable	Yes	< 0.1 bpp
	Puteaux and Puech [30] (1)	RRBE	Separable	No	1 bpp
	Puteaux and Puech [30] (2)	RRBE	Separable	Yes	< 1 bpp
2019	Ge *et al.* [47]	VRAE	Separable	Yes	< 1 bpp

with this problem by introducing a modified version, but it preserves correlations between the original image and its corresponding encrypted version. Theoretically, this makes Xiong *et al.*'s method vulnerable to cryptanalysis.

It is important to note that the use of Paillier or post-quantic cryptosystems implies an increase in the size of the image after encryption. Depending on the method used, if the pixels of the original image are encoded by using 8 bits, they can correspond to nearly 2048 bits in the encrypted domain, as highlighted by Ke *et al.* [66].

4. Comparisons and discussion

Table 1 compares state-of-the-art methods described in Section 3. More specifically, the methods are classified based on the following: year (from 2008 to 2019), the approach used during encoding (RRBE or VRAE), the type of decoding (joint or separable), reversibility, as well the achieved payload. First, we note that the pioneering methods all have the same characteristics, *i.e.* using a VRAE approach during the encoding phase, joint decoding, impossible to reconstruct the original image without error, and a very low payload (< 0.1 bpp). The first separable method was proposed by Zhang in 2012 [33]. We note that this property gives a more concrete application to RDHEI methods. Since 2013 and the method of Ma *et al.* [27], more and more methods opt for a VRAE approach for the encoding phase. This has resulted in a higher payload value (> 0.1 bpp), but still relatively low (< 0.5 bpp). Over the years, although many methods are proposed to achieve full reversibility in the reconstruction phase of the original image, only a few methods can achieve a higher capacity, *i.e.* a payload close to or greater than 0.1 bpp, such as [30]. Furthermore, in Table 1, we indicate the methods which are based on public-key encryption by using a star ($*$). For these methods, their payload is expressed in bits-per-bit of the encrypted image (bpb) instead of bits-per-pixel (bpp). This unit is adopted because, due to the size expansion (Section 3.5), a direct comparison of the payload expressed in bpp is unsuitable and would lead to misinterpretation of the results.

5. The next 12 years and beyond

5.1. High capacity RDHEI methods

In the last few years, several high capacity RDHEI methods have emerged. Indeed, in 2018, the method of Puteaux and Puech has taken the opposite line of all other modern state-of-the-art methods and achieves very good results, with a payload value equal (or very close) to 1 *bpp* and a perfect reversibility. As a result, many new methods have subsequently been inspired by this approach and have further increased payload. Puyang *et al.* proposed the first extension of the EPE-HCRDH approach [67]. They suggested using a more efficient predictor, in order to limit the number of prediction errors. Moreover, they explain that the first and second MSB can be used for data hiding. By using the MSB, an average payload of 1.35 *bpp* is obtained. Yi *et al.* designed a method based on the labeling of a parametric binary tree, where the spatial correlation between pixels in the clear domain is maintained in the encrypted domain within small blocks [68]. Selected pixels are used as reference values to compute the prediction errors which are then highlighted through the labeling of a parametric binary tree. To realize message embedding, the bits of a secret message are inserted by substitution while exploiting the spatial correlation between pixels. Using this process, the authors can reach a payload value in the order of 2 *bpp*. Wu *et al.* also improved the work of Yi *et al.* by reserving room before encryption [69]. In their method [70], Chen and Chang perform a block rearrangement of the MSB planes. This allows them to transform the MSB planes of the original image into a binary sequence that can be compressed by using extended run-length encoding. With this rearrangement and compression, they can free up a large amount of space for data hiding. During the decoding phase, the embedded secret message can be extracted directly from the encrypted image with the help of the data hiding key and the clear marked image can be obtained with the encryption key. Some recent methods have proposed to recursively process all the bitplanes of an image, starting from the MSB plane [71, 72]. In this way, the high correlation between pixels in the clear domain is fully exploited and a payload of the order of 2.4 *bpp* can be achieved. Recently, there have been methods that focused on lossless compression of all the bitplanes of an image and they obtained a payload of more than 3 *bpp* [73, 74]. In addition, current research is aimed at developing a high capacity RDHEI method based on exploiting the homomorphic properties of public key encryption approaches.

5.2. Extensions for other media

In this survey, we have focused exclusively on RDHEI methods applied to uncompressed digital images. Accordingly, it is important to note that some extensions are emerging for 2D vector graphics [75], compressed images [34], video [76] and 3D objects [77]. These formats make encryption and data hiding more challenging due to format compliance and size preservation necessities.

In fact, some RDHEI methods are proposed uniquely for JPEG. In 2014, Qian *et al.* proposed to encode a secret message using Error Correction Codes (ECC) [34]. Then, the same authors suggested to form a new JPEG bitstream with some blocks from the original image [78], where the unused blocks are hidden in the JPEG header by using the same method as in JPEG XT [79]. Chang *et al.* proposed to reserve space for secret message embedding before bitstream encryption [80]. Recently, Qian *et al.* improved their previous scheme [78] using a combination of code mapping and ordered embedding [6]. However, none of these methods allow us to achieve a large payload, even when using a high quality factor. Another issue lies in the fact that most of them are not fully JPEG format compliant. These drawbacks are important problems to be addressed in the future.

In addition, for video, Xu *et al.* designed a RDH method in encrypted H.264/AVC video streams [76]. During H.264/AVC encoding, they proposed to encrypt intra-prediction mode (IPM), motion vector difference (MVD), and signs of residue coefficients. Then, a secret message is embedded into the encrypted H.264/AVC video using histogram shifting. During the decoding phase, data

extraction can be carried out either in the encrypted domain or in the clear domain (*i.e* from the marked reconstructed bitstream) and without any errors. In another method for H.264/AVC, Yao *et al.* used a similar encoding scheme, but also analyzed the inter-frame distortion drift introduced by data hiding [81]. Furthermore, Long *et al.* proposed an approach for RDH in encrypted HEVC video streams [82]. During the encoding phase, the signs and amplitudes of motion vector differences as well as the signs of residual coefficients are encrypted using RC4. The secret message is then embedded into non-zero AC residual coefficients. Moreover, Tew *et al.* [83] proposed a separable authentication scheme for encrypted HEVC video. Syntax elements in HEVC are divided into two groups, where the first group is manipulated to embed authentication data, while the second group is encrypted to mask the video. In their work, authentication can take place in both the encrypted or clear domain, hence separately.

Only a few RDH methods are developed for encrypted 3D objects, despite their wide range of applications [77, 84, 85]. Among them, Jiang *et al.* map vertex coordinates of 3D meshes to integers using scaling and quantization [77]. After this pre-processing stage, additional data is embedded by flipping several LSBs of encrypted coordinates. During the decoding phase, data extraction and original mesh reconstruction are jointly performed by using a smoothing measure function. Although data can be hidden into 3D encrypted objects, the performances of this method are quite poor, including small payload, bad quality of the reconstructed mesh and errors during secret message extraction. Also of interest, Shah *et al.* designed a method based on the use of the Paillier cryptosystem [84, 26]. Even if it is suitable for cloud data management, it is not practical due to a large size expansion after mesh encryption. Contrastingly, Yin *et al.* proposed to embed a secret message by slightly modifying the mesh vertices without changing the mesh topology [85]. However, the main drawback of this method is that the secret message cannot be preserved in the mesh after decryption. All in all, even if some recent papers proposed to embed a secret message into 3D encrypted objects, there is room for further improvement.

Finally, we anticipate future proposals for RDHEI/V where the image/video is encoded in new formats, such as JPEG XL and VVC that support high dynamic range and many other features.

5.3. Security evaluation

Even if some metrics can be used to evaluate the visual security level of encrypted images, there are still a lack of tools to perform the security evaluation of RDHEI methods. In addition to the metrics presented in Section 2.3, some perceptual metrics can be used to assess image confidentiality after encryption [86, 87]. Furthermore, Preishuber *et al.* stated that passing classical statistical tests and those of the NIST test suite [88] are a necessary condition, but not sufficient enough to prove that an encryption scheme used in a RDHEI method is secure [40]. Indeed, they do not reflect attackers that use knowledge of the encryption algorithm during their attack. Attack scenarios such as those employed in cryptanalysis have then to be considered. Note that some non-traditional forms of cryptanalysis of the marked encrypted content which is generated by RDHEI, such as obtaining a sketch [14] of the plain image instead of getting the encryption/decryption key, should also be taken into account for security evaluation. Finally, to the best of our knowledge, there are no methods designed to observe whether the secret message embedded in the encrypted domain is truly undetectable from a steganalysis point of view.

6. Conclusion

In this survey, we present the birth and evolution of RDHEI methods over the last 12 years. First, we detailed the motivations and applications of RDHEI methods. Then, the classes and characteristics of these approaches are presented. The state-of-the-art methods are then explained. With the aid of a comparison table of the representative methods, we are able to see that there is a clear

trade-off between payload and reconstructed image quality. Indeed, only a few methods are able to obtain both a large payload and a perfect reconstruction of an original image. After that, we have discussed emerging applications and open research topics relevant to RDHEI for the next 12 years and beyond.

Over time, RDHEI has become a hot topic whilst still being an emerging technology. The research focusing on theory, framework, methodology and applications needs to be further investigated and developed thoroughly. Indeed, there is a real challenge to obtain the best possible trade-off between payload, reconstructed image quality and security level. Furthermore, specific methods have to be designed for new formats / containers, in particular extensions for images of the JPEG family.

The digital image is certainly worth more than a thousand words. However, recent advancement in Generative Adversarial Network (GAN) further complicates the situation [89, 90]. In addition, privacy and piracy issues will persist for the foreseeable future. These problems will continue to drive research, development and innovation of RDHEI, or more generally the unification of data hiding and encryption, to protect images. How much data can be hidden by trading off the quality of the reconstructed image, robustness of the hidden data, and properties such as separable and commutative is the question that needs to be explored.

References

[1] S. Aslam, Facebook by the numbers: Stats, demographics & fun facts, `https://www.omnicoreagency.com/facebook-statistics/`, 2020.
[2] Y. Tew, K. Wong, An overview of information hiding in H.264/AVC compressed video, IEEE Transactions on Circuits and Systems for Video Technology 24 (2014) 305 – 319.
[3] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital watermarking and steganography, Morgan Kaufmann, 2007.
[4] Y. Tew, K. Wong, R. C.-W. Phan, K. N. Ngan, Multi-layer authentication scheme for hevc video based on embedded statistics, Journal of Visual Communication and Image Representation 40 (2016) 502 – 515.
[5] H. Ren, W. Lu, B. Chen, Reversible data hiding in encrypted binary images by pixel prediction, Signal Processing 165 (2019) 268 – 277.
[6] Z. Qian, H. Xu, X. Luo, X. Zhang, New framework of reversible data hiding in encrypted JPEG bitstreams, IEEE Transactions on Circuits and Systems for Video Technology 29 (2019) 351–362.
[7] S. Xiang, X. Luo, Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group, IEEE Transactions on Circuits and Systems for Video Technology 28 (2018) 3099–3110.
[8] H. Ge, Y. Chen, Z. Qian, J. Wang, A high capacity multi-level approach for reversible data hiding in encrypted images, IEEE Transactions on Circuits and Systems for Video Technology 29 (2019) 2285–2295.
[9] J. He, J. Chen, W. Luo, S. Tang, J. Huang, A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams, IEEE Transactions on Circuits and Systems for Video Technology 29 (2018) 3501–3515.
[10] B. Chen, X. Wu, W. Lu, H. Ren, Reversible data hiding in encrypted images with additive and multiplicative public-key homomorphism, Signal Processing 164 (2019) 48–57.
[11] C. E. Shannon, Communication theory of secrecy systems, The Bell System Technical Journal 28 (1949) 656–715.
[12] D. Coppersmith, D. B. Johnson, S. M. Matyas, A proposed mode for triple-DES encryption, IBM Journal of Research and Development 40 (1996) 253–262.
[13] J. Daemen, V. Rijmen, The design of Rijndael: AES –the Advanced Encryption Standard, Springer–Verlag, Berlin, 2002.
[14] K. Minemura, K. Wong, R. Phan, K. Tanaka, A novel sketch attack for H.264/AVC format-compliant encrypted video, IEEE Transactions on Circuits and Systems for Video Technology 27 (2017) 2309 – 2321.
[15] W. Puech, M. Chaumont, O. Strauss, A reversible data hiding method for encrypted images, in: Security, Forensics, Steganography, and Watermarking of Multimedia Contents, volume X, International Society for Optics and Photonics, 2008, pp. 68191E–68191E.
[16] G. S. Poh, K. M. Martin, An efficient buyer-seller watermarking protocol based on chameleon encryption, in: International Workshop on Digital Watermarking (IWDW), 2009, pp. 433–447.
[17] D. Kundur, K. Karthik, Video fingerprinting and encryption principles for digital rights management, Proceedings of the IEEE 92 (2004) 918 – 932.
[18] M. S. A. Karim, K. Wong, Universal data embedding in encrypted domain, Signal Processing 94 (2014) 174 – 182.
[19] K. S. Wong, K. Tanaka, Data embedding for geo-tagging any contents in smart device, in: Region 10 Symposium, 2014 IEEE, 2014, pp. 527–530.
[20] H. Sohn, W. D. Neve, Y. M. Ro, Privacy protection in video surveillance systems: Analysis of subband-adaptive scrambling in JPEG XR, IEEE Transactions on Circuits and Systems for Video Technology 21 (2011) 170–177.
[21] T. Hey, S. Tansley, K. Tolle (Eds.), The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, Redmond, Washington, 2009.
[22] S. Ong, Data Insertion and Scrambling for Unified Scalable Information Hiding, Ph.D. thesis, University of Malaya, 2015.
[23] S. Ong, K. Wong, K. Tanaka, A scalable reversible data embedding method with progressive quality degradation functionality, Signal Processing: Image Communication 29 (2014) 135–149.
[24] S. Ong, K. Minemura, K. Wong, Progressive quality degradation in JPEG compressed image using DC block orientation with rewritable data embedding functionality, in: IEEE International Conference on Image Processing (ICIP), 2013, pp. 4574 – 4578.
[25] S. Ong, K. Wong, K. Tanaka, Scrambling-embedding for JPEG compressed image, Signal Processing 109 (2015) 38 – 53.
[26] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), Springer, 1999, pp. 223–238.
[27] K. Ma, W. Zhang, X. Zhao, N. Yu, F. Li, Reversible data hiding in encrypted images by reserving room before encryption, IEEE Transactions on Information Forensics and Security 8 (2013) 553–562.
[28] X. Cao, L. Du, X. Wei, D. Meng, X. Guo, High capacity reversible data hiding in encrypted images by patch-level sparse representation, IEEE Transactions on Cybernetics 46 (2016) 1132–1143.
[29] X. Zhang, J. Long, Z. Wang, H. Cheng, Lossless and reversible data hiding in encrypted images with public-key cryptography, IEEE Transactions on Circuits and Systems for Video Technology 26 (2016) 1622–1631.

[30] P. Puteaux, W. Puech, An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images, IEEE Transactions on Information Forensics and Security 13 (2018) 1670–1681.
[31] W. Hong, T.-S. Chen, H.-Y. Wu, An improved reversible data hiding in encrypted images using side match, IEEE Signal Processing Letters 19 (2012) 199–202.
[32] X. Zhang, Reversible data hiding in encrypted image, IEEE Signal Processing Letters 18 (2011) 255–258.
[33] X. Zhang, Separable reversible data hiding in encrypted image, IEEE Transactions on Information Forensics and Security 7 (2012) 826–832.
[34] Z. Qian, X. Zhang, S. Wang, Reversible data hiding in encrypted JPEG bitstream, IEEE Transactions on Multimedia 16 (2014) 1486–1491.
[35] S. Li, G. Chen, X. Mou, On the dynamical degradation of digital piecewise linear chaotic maps, International Journal of Bifurcation and Chaos 15 (2005) 3119–3151.
[36] R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM 21 (1978) 120–126.
[37] F. Huang, J. Huang, Y.-Q. Shi, New framework for reversible data hiding in encrypted domain, IEEE Transactions on Information Forensics and Security 11 (2016) 2777–2789.
[38] R. M. Rad, K. Wong, J. Guo, A unified data embedding and scrambling method, IEEE Transactions on Image Processing 23 (2014) 1463–1475.
[39] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (2004) 600–612.
[40] M. Preishuber, T. Hütter, S. Katzenbeisser, A. Uhl, Depreciating motivation and empirical security analysis of chaos-based image and video encryption, IEEE Transactions on Information Forensics and Security 13 (2018) 2137–2150.
[41] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948) 379–423.
[42] Y. Wu, J. P. Noonan, S. Agaian, et al., NPCR and UACI randomness tests for image encryption, Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications 1 (2011) 31–38.
[43] W. Zhang, K. Ma, N. Yu, Reversibility improved data hiding in encrypted images, Signal Processing 94 (2014) 118–127.
[44] S. Ong, K. Wong, X. Qi, K. Tanaka, Beyond format-compliant encryption for JPEG image, Signal Processing: Image Communication 31 (2015) 47 – 60.
[45] D. Xiao, Y. Xiang, H. Zheng, Y. Wang, Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism, Journal of Visual Communication and Image Representation 45 (2017) 1–10.
[46] D. Xu, R. Wang, Separable and error-free reversible data hiding in encrypted images, Signal Processing 123 (2016) 9–21.
[47] H. Ge, Y. Chen, Z. Qian, J. Wang, A high capacity multi-level approach for reversible data hiding in encrypted images, IEEE Transactions on Circuits and Systems for Video Technology 29 (2019) 2285–2295.
[48] X. Li, J. Li, B. Li, B. Yang, High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion, Signal Processing 93 (2013) 198–205.
[49] M. Long, Y. Zhao, X. Zhang, F. Peng, A separable reversible data hiding scheme for encrypted images based on Tromino scrambling and adaptive pixel value ordering, Signal Processing 176 (2020) 107703.
[50] M. S. A. Karim, K. Wong, Data embedding in random domain, Signal Processing 108 (2015) 56 – 68.
[51] Z. Qian, X. Zhang, Reversible data hiding in encrypted images with distributed source encoding, IEEE Transactions on Circuits and Systems for Video Technology 26 (2016) 636–646.
[52] D. Slepian, J. Wolf, Noiseless coding of correlated information sources, IEEE Transactions on Information Theory 19 (1973) 471–480.
[53] W. Liu, W. Zeng, L. Dong, Q. Yao, Efficient compression of encrypted grayscale images, IEEE Transactions on Image Processing 19 (2009) 1097–1102.
[54] X. Wu, W. Sun, High-capacity reversible data hiding in encrypted images by prediction error, Signal Processing 104 (2014) 387–400.
[55] I. C. Dragoi, H.-G. Coanda, D. Coltuc, Improved reversible data hiding in encrypted images based on reserving room after encryption and pixel prediction, in: European Signal Processing Conference (EUSIPCO), 2017, pp. 2186–2190.
[56] I. C. Dragoi, D. Coltuc, Reversible data hiding in encrypted images based on reserving room after encryption and multiple predictors, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 2102–2105.
[57] Y.-C. Chen, C.-W. Shiu, G. Horng, Encrypted signal-based reversible data hiding with public key cryptosystem, Journal of Visual Communication and Image Representation 25 (2014) 1164–1170.
[58] Y. Ke, M. Zhang, J. Liu, Separable multiple bits reversible data hiding in encrypted domain, in: International Workshop on Digital Watermarking (IWDW), Springer, 2016, pp. 470–484.
[59] C.-W. Shiu, Y.-C. Chen, W. Hong, Encrypted image-based reversible data hiding with public key cryptography from difference expansion, Signal Processing: Image Communication 39 (2015) 226–233.
[60] H.-T. Wu, Y.-M. Cheung, J. Huang, Reversible data hiding in Paillier cryptosystem, Journal of Visual Communication and Image Representation 40 (2016) 765–771.
[61] M. Li, Y. Li, Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding, Signal Processing 130 (2017) 190–196.
[62] S. Xiang, X. Luo, Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group, IEEE Transactions on Circuits and Systems for Video Technology 28 (2018) 3099–3110.
[63] S. Zheng, Y. Wang, D. Hu, Lossless data hiding based on homomorphic cryptosystem, IEEE Transactions on Dependable and Secure Computing (2019) 1–1, *doi: 10.1109/TDSC.2019.2913422*.
[64] C. Jiang, Y. Pang, Encrypted images-based reversible data hiding in Paillier cryptosystem, Multimedia Tools and Applications 79 (2020) 693–711.
[65] L. Xiong, D. Dong, Z. Xia, X. Chen, High-capacity reversible data hiding for encrypted multimedia data with somewhat homomorphic encryption, IEEE Access 6 (2018) 60635–60644.
[66] Y. Ke, M. Zhang, J. Liu, T. Su, X. Yang, A multilevel reversible data hiding scheme in encrypted domain based on LWE, Journal of Visual Communication and Image Representation 54 (2018) 133–144.
[67] Y. Puyang, Z. Yin, Z. Qian, Reversible data hiding in encrypted images with two-MSB prediction, in: IEEE Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7.
[68] S. Yi, Y. Zhou, Separable and reversible data hiding in encrypted images using parametric binary tree labeling, IEEE Transactions on Multimedia 21 (2019) 51–64.
[69] Y. Wu, Y. Xiang, Y. Guo, J. Tang, Z. Yin, An improved reversible data hiding in encrypted images using parametric binary tree labeling, IEEE Transactions on Multimedia 22 (2020) 1929 – 1938.
[70] K. Chen, C.-C. Chang, High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement, Journal of Visual Communication and Image Representation 58 (2019) 334–344.
[71] P. Puteaux, W. Puech, EPE-based huge-capacity reversible data hiding in encrypted images, in: IEEE Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7.
[72] P. Puteaux, W. Puech, A recursive reversible data hiding in encrypted images method with a very high capacity, IEEE Transactions on Multimedia (2020) 1–1, *doi: 10.1109/TMM.2020.2985537*.
[73] Z. Yin, Y. Xiang, X. Zhang, Reversible data hiding in encrypted images based on multi-MSB prediction and Huffman coding, IEEE Transactions on Multimedia 22 (2020) 874 – 884.

[74] Z. Yin, Y. Peng, Y. Xiang, Reversible data hiding in encrypted images based on pixel prediction and bit-plane compression, IEEE Transactions on Dependable and Secure Computing (2020) 1–1, *doi: 10.1109/TDSC.2020.3019490*.
[75] F. Peng, W.-Y. Jiang, Y. Qi, Z.-X. Lin, M. Long, Separable robust reversible watermarking in encrypted 2D vector graphics, IEEE Transactions on Circuits and Systems for Video Technology (2020).
[76] D. Xu, R. Wang, Y. Q. Shi, Reversible data hiding in encrypted H. 264/AVC video streams, in: International Workshop on Digital Watermarking (IWDW), Springer, 2013, pp. 141–152.
[77] R. Jiang, H. Zhou, W. Zhang, N. Yu, Reversible data hiding in encrypted three-dimensional mesh models, IEEE Transactions on Multimedia 20 (2017) 55–67.
[78] Z. Qian, H. Zhou, X. Zhang, W. Zhang, Separable reversible data hiding in encrypted JPEG bitstreams, IEEE Transactions on Dependable and Secure Computing 15 (2016) 1055–1067.
[79] T. Richter, A. Artusi, T. Ebrahimi, JPEG XT: A new family of JPEG backward-compatible standards, IEEE Multimedia 23 (2016) 80–88.
[80] J.-C. Chang, Y.-Z. Lu, H.-L. Wu, A separable reversible data hiding scheme for encrypted JPEG bitstreams, Signal Processing 133 (2017) 135–143.
[81] Y. Yao, W. Zhang, N. Yu, Inter-frame distortion drift analysis for reversible data hiding in encrypted H. 264/AVC video bitstreams, Signal Processing 128 (2016) 531–545.
[82] M. Long, F. Peng, H.-y. Li, Separable reversible data hiding and encryption for HEVC video, Journal of Real-Time Image Processing 14 (2018) 171–182.
[83] Y. Tew, K. Wong, R. C. Phan, K. N. Ngan, Separable authentication in encrypted HEVC video, Multimedia Tools and Applications 77 (2018) 24165–24184.
[84] M. Shah, W. Zhang, H. Hu, H. Zhou, T. Mahmood, Homomorphic encryption-based reversible data hiding for 3D mesh models, Arabian Journal for Science and Engineering 43 (2018) 8145–8157.
[85] Z. Yin, N. Xu, F. Wang, Separable reversible data hiding based on integer mapping and multi-MSB prediction for encrypted 3D mesh models, arXiv (2019) arXiv–1908.
[86] N. Le Philippe, V. Itier, W. Puech, Visual saliency-based confidentiality metric for selective crypto-compressed JPEG images, in: IEEE International Conference on Image Processing (ICIP), 2017, pp. 4347–4351.
[87] S. Guo, T. Xiang, X. Li, Y. Yang, PEID: A perceptually encrypted image database for visual security evaluation, IEEE Transactions on Information Forensics and Security 15 (2019) 1151–1163.
[88] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, A statistical test suite for random and pseudorandom number generators for cryptographic applications, Technical Report, Gaithersburg, MD, USA, 2001.
[89] I. Korshunova, W. Shi, J. Dambre, L. Theis, Fast face-swap using convolutional neural networks, in: IEEE International Conference on Computer Vision (ICCV), 2017, p. 3677–3685.
[90] Y. Nirkin, I. Masi, A. T. Tuan, T. Hassner, G. Medioni, On face segmentation, face swapping, and face perception, in: IEEE International Conference on Automatic Face & Gesture Recognition (FG), 2018, p. 98–105.