Multimed Tools Appl
https://doi.org/10.1007/s11042-017-5580-2

Cloud based image color transfer and storage in encrypted domain

Amitesh Singh Rajput[1] · **Balasubramanian Raman**[1]

Received: 31 May 2017 / Revised: 4 November 2017 / Accepted: 26 December 2017

Abstract Cloud infrastructures are developed and maintained by third parties and users are always concerned about processing and storing their data over the cloud. Recent technologies such as high definition and 360-degree images/videos require efficient color processing, and current trend towards the cloud computing has initiated a necessity of performing color transfer remotely by untrusted third party servers. Nowadays, this field is emerging fast due to its inherent potential and research work in this direction is highly demanded. To address this necessity, we present a system that addresses the challenge of performing privacy preserving color transfer over third party servers. We use a one-dimensional chaotic logistic map coupled with ramp secret sharing scheme in a manner that secret images can be stored and processed for color transfer in the encrypted domain. Experimental results and security analysis demonstrate effectiveness of the approach against existing techniques of color transfer as well as image encryption.

Keywords Color transfer · Encrypted domain processing · Cloud computing

1 Introduction

Cloud computing provides cost effective pay-as-you-go services with business continuity. Users are relieved from infrastructure maintenance responsibilities and can accomplish the services directly offered by cloud service providers at reasonable costs. Consequently, new cloud enabled gadgets are evolving to store and process user data over the cloud including Chromebook from big software giants such as Google, HP, Samsung etc. These gadgets possess only small storage space using flash drive and rest of the data is stored and processed

✉ Amitesh Singh Rajput
asr88.dcs2015@iitr.ac.in

[1] Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India

over cloud servers. Regardless of these benefits, cloud computing has some inherent weaknesses: cloud service providers are always assumed untrusting and may possess malicious motivations. While cloud infrastructures are developed and maintained by third parties, users concern more regarding their data and such situation poses a challenging problem for realizing data storage and processing over the cloud. Data confidentiality is among major concerns when moving data to the cloud [38].

In today's world, majority of the data is shared in the form of digital images and techniques to store and process them in a secure manner over cloud servers are required. Existing encryption methods [6, 10, 12, 22, 30, 34] are not suited for processing the images in encrypted form and may distort the contents when the processed image is decrypted. For example, suppose we want to perform the task of mean filtering in Encrypted Domain (ED) with a mask of size 3×3. The mean filter simply computes mean of the pixel intensities in the filter mask and replaces the center pixel value with the computed one. The filter works fine in Plain Domain (PD) however, image pixel values are modified during encryption (due to integration of different techniques such as diffusion, substitution, XOR operations, etc.) and computation of mean filtering over the modified pixels results in distorted values of the processed image which may become inadequate after the decryption. Moreover, image steganography [4, 5, 7, 11, 13, 15] has been widely used for secure transmission of the secret data. However, the image need to be decoded over the cloud server to process the data. Hence, new state-of-the-art requirements impose the use of homomorphic encryption mechanisms such that secret images can be processed in encrypted form over the cloud. This paper presents a novel approach for privacy preserving color transfer such that secret images can be processed without revealing any information at untrusted cloud infrastructures. Additionally, a large number of images can be stored at reasonable costs without worrying about privacy issues over the large storage pools provided by cloud service providers using the proposed approach.

During the past few years, various schemes have been proposed to process secret images in ED. A face search scheme in ED is proposed in [37], where a novel method is used to extract and locate face object region within the encrypted image. Initially, as part of preprocessing, the image is transformed to DCT (Discrete Cosine Transform) domain and then coefficients are utilized for encryption and decryption. A novel scheme for ED facial expression recognition is presented in [24]. The authors used Local Fisher Discriminant Analysis (LFDA) to hide human subject's facial images from untrusted entities and homomorphic properties of Paillier cryptosystem are utilized well over the server. Since LFDA is robust and linear operations such as normalization and projection are involved in the classification step, ED facial expression recognition is performed proficiently. An image scaling and cropping scheme is proposed in [20]. The scheme uses Shamir's secret sharing scheme to generate image shares and facilitates cropping and scaling of the secret image in ED at cloud data centres (CDCs). Feature extraction using privacy preserving SIFT in ED is proposed in [14]. SIFT is a well known method for detecting local features of an image and homomorphic encryption based secure SIFT is proposed by the authors. Paillier cryptosystem is used as the platform for designing secure SIFT and results are discussed and compared to the original one. Spatial domain image enhancement operations in ED are presented in [16]. During pre-processing phase, non-terminating division sequences are converted to terminating ones and then further processed over cloud data centers. In this paper, we propose a novel approach using HE for efficient color transfer and storage of secret images over third party cloud servers.

In the contemporary world of cloud computing, offering only storage seems to be a generic job which is available with almost every cloud service provider. Moreover, the

technology is moving towards high definition/360-degree images and videos, and offering privacy preserving services related to processing as-well-as storage forms an additional advantage. Previous ED schemes in the literature [14, 16, 20, 24, 37] consist of image processing operations which can be performed in PD and then images can be encrypted and stored over the cloud. However, every task is unique and enables the cloud service provider to get additional benefit of add-on services rather than storage. Appropriately, following the same perception as of the existing schemes with a different problem space, we perform the task of color transfer in ED over the cloud with storage as an implicit feature provided by cloud service providers.

1.1 Color transfer

Maintaining persistent colors across panoramic images/videos by multi-camera arrays is emerging as consequential requirement these days. Color transfer is used to correct non-uniformity between images of different colors by removing dominant and undesirable color shed and modifying the colors. Previously, color transfer has received less interest due to other techniques of image stitching and blending [36]. However, the growing demand of high definition and 360-degree images/videos has led the people to distinguish that only blending cannot remove all color differences and hence, good color transfer techniques are required. A variety of color correction approaches exist including gain compensation [21, 32], color balancing [29] and color transfer [23, 26, 35].

1.1.1 Color transfer between images, Reinhard et al. [26]

A pioneering work for transferring colors between images was proposed by Reinhard et al. [26]. Due to high correlation between pixels of the RGB (Red, Green and Blue) color channels of the image, pixel values are transformed to $l\alpha\beta$ (achromatic, chromatic yellow-blue and red-green) color space. The color channels are then modified separately and returned to RGB color space after processing. The method proposed by Reinhard et al. [26] transforms image colors in $l\alpha\beta$ color space. Initially, Reinhard's scheme subtracts mean of the target image μ_t from pixel values of the target image such that a difference image T_d can be obtained. Then, product of the fraction of standard deviation scores σ of reference σ_{ref} and target σ_t images is accomplished with T_d ($\sigma \times T_d$). Finally, the resulting product is then added to the mean of the reference image μ_{ref} to get the color transformed image. Let the matrix depicted in Fig. 1a represents the actual pixel values of the red color channel of the target image. Considering $\mu_{ref} = 68$ and $\sigma = 1.8991$ with $\mu_t = 50.5$, a working example of color transfer using Reinhard's scheme in RGB color space is demonstrated in Fig. 1. For convenience in understanding, only red color channel is considered and transformation of actual pixel values of the target image is demonstrated. In this paper, we map the linear statistical transformations proposed by Reinhard et al. [26] over the separated RGB color space (due to image encryption). Advantage includes privacy preserving transformation of color palette of the target image as per the reference image.

Applications of color transfer vary from elusive processing on images to advance their appearance for more remarkable modifications such as converting a day light, enhancement of underwater and satellite images etc. For color measurement while capturing the pictures, camera coordinates are calibrated to a standard color space "XYZ" by a mapping procedure. However such mapping from RGB to XYZ may cause errors which can be avoided by using polynomial color transfer (PCC) and a new polynomial-type regression known as root-PCC (RPCC) is proposed in [9]. Another work addressing the requirement of region based

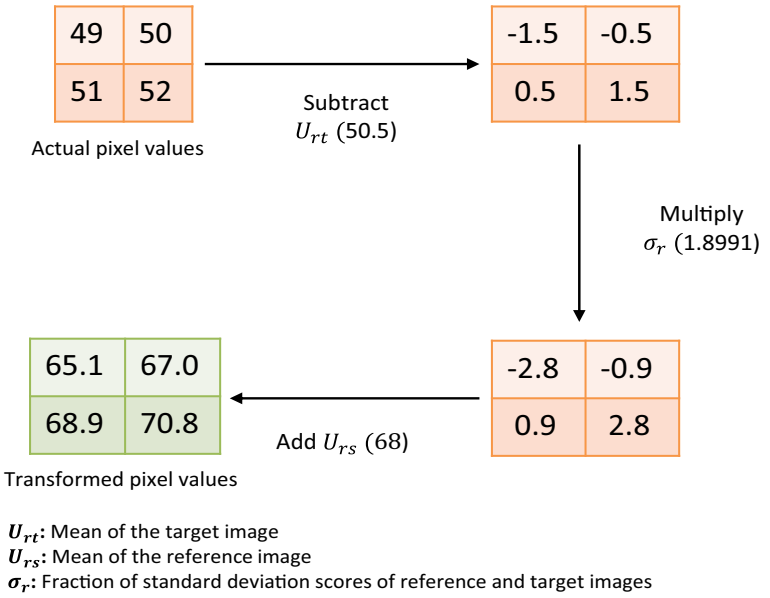


Fig. 1 Working example of Reinhard's scheme [26] in RGB color space

color transfer is presented in [18]. Image segmentation is performed by marker-controlled watershed transformation followed by point feature correspondences for matching features between two images. Results are evaluated and demonstrated with various image sets. Automatic color transfer is used to enhance underwater images in [17]. Since color distortion and scattering are two major problems with underwater images, trigonometric bilateral filters are well used.

While cloud infrastructures provide on demand provisioning of resources and several operations related to image enhancement are evolving in ED over the cloud, a need to perform privacy preserving color transfer also arises. Imagine a scenario where users can send their personal images to the cloud without worrying about processing/storage requirements and can retrieve the color enhanced images on demand at anytime and anywhere. The scenario becomes more interesting when images are stored and processed over cloud in encrypted form such that image information cannot be revealed over third party servers. Since well established methods exist for color transfer between images in PD [9, 17, 18, 21, 23, 26, 29, 32, 35], the works yet not reached a satisfactory level to be deployed directly in ED. Transforming colors between encrypted images when image contents are distorted (due to encryption) is a challenging problem and motivated us to devise the proposed approach. Novelty of the proposed approach lies in transforming color palette of the encrypted images. The two main contributions of our work can be summarized as follows:

1. We integrate Ramp secret sharing scheme with pixel permutation such that obfuscated image shares are generated and no secret image information is revealed while processing over third party cloud servers. Color transfer consists of a set of primitive operations such as addition/subtraction and scalar multiplication. Therefore, integration of Ramp secret sharing and pixel permutation for color transfer is a challenging task. We address this challenge by modifying the linear color transformations proposed by Reinhard et al. [26] such that colors transfer can be realized in ED for secret images of the cloud user.

2. Unlike the state-of-art Shamir's secret sharing scheme, we use Ramp secret sharing which is extended version of the Shamir's scheme. As a result, the size of image shares is drastically reduced making the task of image storage more convenient over multiple cloud servers.

The subsequent part of this paper is organized as follows: Section 2 presents the preliminaries required to understand the proposed approach with detailed description of ramp secret sharing and formulation of one dimensional chaotic map. The proposed image encryption and color transfer algorithms are presented in Section 3. Sections 4 and 5 provide analysis of the experiments done for verification and security purpose, respectively. Finally, Section 6 concludes the paper and discusses future prospects of the proposed approach.

2 Preliminaries

2.1 Ramp secret sharing

Ramp secret sharing is extended from Shamir's secret sharing scheme (SSS), a secret sharing scheme known for additive and multiplicative homomorphism. SSS was proposed by Adi Shamir [27] in 1979. Given a secret s SSS divides it into n shares such that minimum $t (t \leq n)$ shares are required to reconstruct the secret back. A polynomial function (with $t-1$ degree) is used to generate the secret shares, described as follows -

$$f(x) = (c_0 + c_1 x + c_2 x^2 + \cdots + c_{t-1} x^{t-1}) mod\ p \tag{1}$$

where p and $c_0, c_1, \cdots c_{t-1}$ are prime numbers and coefficients of the polynomial, respectively. The secret is assigned to the very first coefficient c_0, whereas others $(c_1, c_2, \cdots c_{t-1})$ are selected randomly in the range $(0, p)$. During the past few years, various works [16, 25] have been proposed in the literature using SSS for privacy preserving distribution and processing of secret images in ED. Specially, the work proposed by Rajput et al. [25] is focused to transfer image colors in ED. However, these works suffer from the drawbacks of size expansion and collusion attack, discussed as follows -

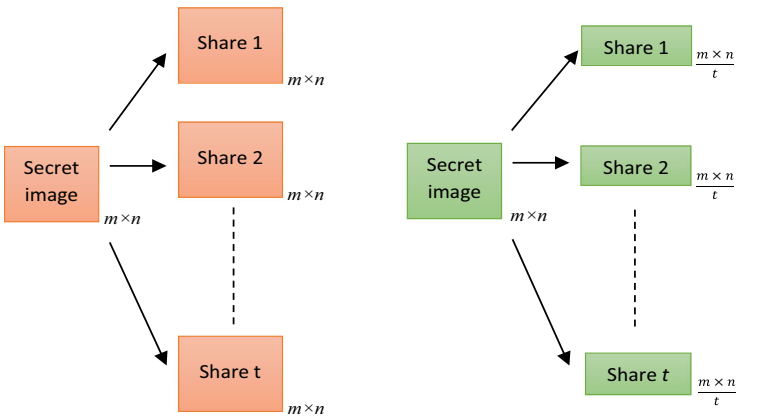
Size expansion Using SSS, n shares are generated and transmitted to different cloud servers. Each pixel of the secret image is selected and its corresponding shares are generated using the polynomial function depicted in (1). The same process continues till all pixels of the secret image are processed. As a result, each image share is also of the same size as of secret image. Consequently, if total number of shares is q, then storage and distribution overhead is increased by q times (as compared to the original secret image).

Collusion attack The schemes are vulnerable due to collusion attack. Collusion attack involves t cloud servers for accessing t image shares. Once t image shares are gathered, the secret image can be retrieved.

On the other hand, ramp secret sharing involves using pixel values of the image as coefficients of the polynomial function depicted in (1). As a result, size of the resulting image shares is drastically reduced. Ramp secret sharing is the extended version of SSS where pixel values of the secret image are used in place of random coefficients. The summation of size of all the shares is equivalent to the size of the actual secret image. Using SSS, the size of each image share is equal to the size of the secret image. As a

consequence, the total network traffic also increases with the number of shares. In our scheme, since Ramp secret sharing scheme is used, total size of all the shares is equal to the size of the secret image or it can be said that size of the secret image is divided into number of shares using Ramp secret sharing scheme. The reader is advised to refer to Fig. 2 for more clarity regarding total size of the image shares where $m \times n$ represents the number of rows and columns of the secret image and t is the total number of image shares.

In the previous ED schemes using SSS scheme, there is a mandatory condition of storing all the shares on different CDCs such that collision attack can be avoided. We use image permutation to resist the collusion attacks and hide image information while processing over the cloud. As a consequence, if any adversary succeeds to obtain the threshold number of shares, a permuted image can be reconstructed and no image information is revealed. The 360-bit secret key is then further required to decrypt the permuted image. A proficient work for ramp secret sharing was presented by Thien and Lin [31] and authors used image pixel values (coupled with image permutation) as coefficients in the polynomial function. However, no processing was performed over the image shares (due to image permutation). The authors used ramp secret sharing only for the purpose of efficient transmission of secret images. In this paper, we use ramp secret sharing used by Thien and Lin [31], for RGB color channels of the secret image. Advantage of using Ramp secret sharing includes reduced share size, however the biggest challenge lies in homomorphic processing as Ramp secret sharing is partial homomorphic to multiplication. Considering this challenge, we modified Reinhard et al.'s [26] scheme in a manner such that colors of the target image (encrypted using Ramp secret sharing) are transformed as per the colors of the reference image in ED, which is the very first work in this field. Additionally, key dependent image permutation is used to resist collusion attacks and preserve confidentiality of image contents at CDCs. The proposed approach is discussed in detail in Section 3.



(a) Shamir's secret sharing scheme (b) Ramp secret sharing scheme

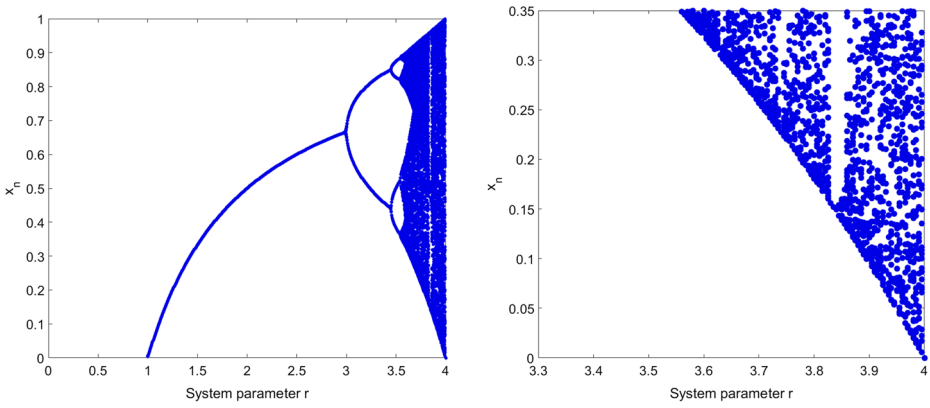
Fig. 2 Total size of image shares using **a** Shamir's secret sharing scheme; **b** Ramp secret sharing scheme

2.2 Chaotic map

Chaotic maps have complicated dynamic behavior and good autocorrelation properties are achieved using logistic mapping sequences. They are mathematical functions, typically used in cryptography for achieving randomness. These functions require initial values to generate chaotic sequences. We use one-dimensional chaotic logistic map to encrypt reference and target images. Instead, other higher dimension logistic maps can also be used for the same purpose. Higher dimensional logistic maps are good option in the condition when encryption algorithm is dependent on pixel modification techniques such as diffusion. We are not using pixel diffusion and only pixel permutation is employed using one-dimensional logistic map. The higher dimensional logistic map takes the computations to higher levels and as we are already using Ramp secret sharing along with image permutation for image encryption, one-dimensional chaotic logistic map is preferred. In future, higher dimensional logistic map can be used instead of using one-dimensional logistic map, only the computations will increase to another level. The one-dimensional chaotic logistic map is defined as -

$$x_{n+1} = r x_n (1 - x_n) \tag{2}$$

where, r is system or control parameter in the interval (0, 4). We have employed chaotic logistic map with different initial values. Bifurcation of the chaotic map is depicted in Fig. 3a where various characteristics of the chaotic behavior for different values of r can be observed. The horizontal and vertical axis illustrate system parameter r as well as possible values of x_n, respectively. Blank window is a common problem with chaotic logistic map and Fig. 3b depicts the scenario that a blank space appears when value of the system parameter is 3.828. In view of the fact that chaotic logistic map has different behaviors for different values of r and a blank space appears at the value of 3.828, we concentrate on the system parameter to be constant at 3.999 in our work (to get resulting chaotic sequences in a uniform manner).



(a) Chaotic logistic map (b) Blank window problem

Fig. 3 Bifurcation of the chaotic logistic map and blank window problem

2.3 Computation of initial values from secret key

Initial values are one of the important constituents to be considered when using chaotic maps. A small modification in the initial value leads to completely different chaotic behavior. Considering this fact, we have computed initial values of the chaotic map from the secret key. Secret key k is defined as a 360-bit binary string classified into nine subsections $\{x_1, x_2, x_3, y_1, y_2, y_3, v_1, v_2, v_3\}$. These binary strings are used to find initial values and iterate chaotic logistic map for different encryption scenarios. The key composition is shown in Fig. 4. Initially, a fraction value f_{s1} is derived from a 52-bit binary string $\{b_{-1}, b_{-2}, ...b_{-52}\}$ considered from first part of the secret key x_1 using IEEE 754 double-precision binary floating point format for the fraction part [34], depicted in (3) as follows -

$$f_{s1} = \sum_{i=1}^{52} b_{-i} 2^{-i} \tag{3}$$

In the same manner, f_{s2} and f_{s3} are computed using x_2 and x_3. The last part of secret key $\{v_1, v_2, v_3\}$ contains 16-bit string each and individually converted to decimal numbers $\{v_1', v_2', v_3'\}$. A set of additional values $\{A_r, A_g, A_b\}$ is derived from three color channels $\{S_r, S_g, S_b\}$ of the secret image S by computing mean (μ) of pixel values of each color channel and taking the fraction part, depicted in (4)–(6).

$$A_r = \mu(S_r) \tag{4}$$

$$A_g = \mu(S_g) \tag{5}$$

$$A_b = \mu(S_b) \tag{6}$$

The set $\{f_{s1}, f_{s2}, f_{s3}, A_r, A_g, A_b, v_1', v_2', v_3'\}$ is then used to compute initial values $\{x_{01}', x_{02}', x_{03}'\}$ of the chaotic map as follows -

$$x_{01}' = (A_r + f_{s1} \times v_1') mod\ 1 \tag{7}$$

$$x_{02}' = (A_g + f_{s2} \times v_2') mod\ 1 \tag{8}$$

$$x_{03}' = (A_b + f_{s3} \times v_3') mod\ 1 \tag{9}$$

Appropriately, another set of initial values $\{y_{01}', y_{02}', y_{03}'\}$ is computed from next part of secret key $\{y_1, y_2, y_3\}$. The initial values $\{x_{01}', x_{02}', x_{03}', y_{01}', y_{02}', y_{03}'\}$ are used to generate random sequences from chaotic map where length of chaotic sequence equals to number of pixels in the image.

3 Privacy preserving color transfer in ED

3.1 Overview

Architecture of the proposed approach is shown in Fig. 5. The proposed approach is designed to run in cloud environment and the system we assume is "honest-but-curious

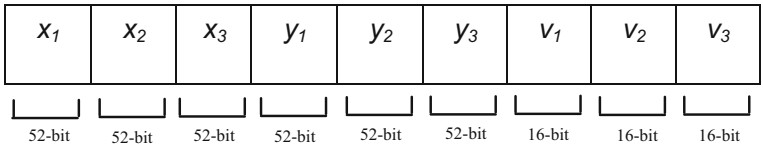


Fig. 4 Key composition

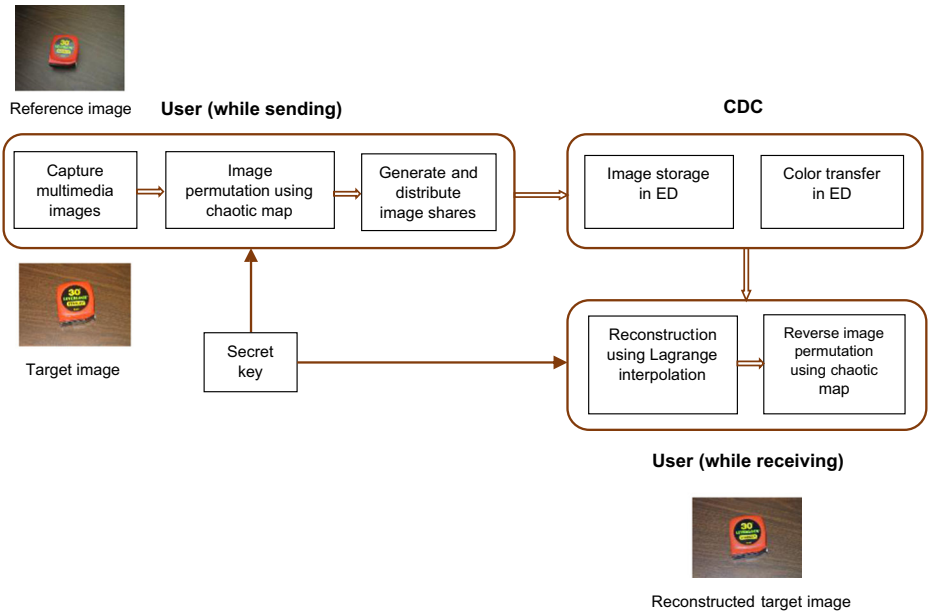


Fig. 5 Architecture of the proposed approach

adversary model" where cloud server performs desired tasks, however is curious about knowing data of the user. Hence, in order to maintain user's privacy, our proposed approach initially encrypts (image permutation and share generation are performed as part of encryption) reference and target images at user end such that privacy preserving storage can be performed over CDCs. On the other hand, only target image is encrypted and transmitted to CDCs in case of color transfer.

3.2 Computation of color transformation values at user end

Image storage consists of simple encryption and decryption of user images. However, in order to perform color transfer operations, it is not straightforward to simply integrate Ramp secret sharing and chaotic map with Reinhard et al.'s [26] scheme directly. In contrast, substantial modifications are required and discussed in this section. Initially, color transformation values are computed at the user end and transmitted to CDCs for efficient color transfer. The RGB color channels of the target image are extracted and color transformation values $\{\sigma_r, \sigma_g, \sigma_b\}$ are computed for each color channel as follows -

$$\sigma_r = \lfloor(\sigma_{rs}/\sigma_{rt})\rceil \tag{10}$$

$$\sigma_g = \lfloor(\sigma_{gs}/\sigma_{gt})\rceil \tag{11}$$

$$\sigma_b = \lfloor(\sigma_{bs}/\sigma_{bt})\rceil \tag{12}$$

where, $\{\sigma_{rs}, \sigma_{gs}, \sigma_{bs}\}$ and $\{\sigma_{rt}, \sigma_{gt}, \sigma_{bt}\}$ represent the standard deviation scores of red, green and blue color channels of the reference and target images, respectively. In order to avoid floating point errors due to modulo operations while processing and reconstruction, the color transformation values are rounded to the nearest integer using a rounding function $\lfloor . \rceil$. The set of color transformation values $\{\sigma_r, \sigma_g, \sigma_b\}$ is then transmitted to all the CDCs

in any manner because the values do not reveal any useful information about secret images. An alternate way is to hide these values in the share itself using any lossless embedding technique. Furthermore, another set of values $\{r', g', b'\}$ is prepared to post-process the image after receiving from the cloud server as follows -

$$r' = r_t \sigma_r - r_s \tag{13}$$

$$g' = g_t \sigma_g - g_s \tag{14}$$

$$b' = b_t \sigma_b - b_s \tag{15}$$

The values represented by r_s, g_s, b_s and r_t, g_t, b_t are mean scores of red, green and blue color channels of reference and target images, respectively.

3.3 Image encryption

The task of image encryption is performed at the user end. We use ramp secret sharing (discussed in Section 2.1) for generating image shares. Problem with ramp secret sharing arises when mapping of image pixels to coefficients of the polynomial function (depicted in (1)) reveals the image information. This can be clearly observed from the image shares shown in Fig. 6c. The image shares are generated directly from secret image shown in Fig. 6a using the polynomial function depicted in (1) with pixel values of the secret image as coefficients. To overcome this problem, pixels of the secret image are permuted before creation of image shares. Image pixels are permuted (position wise) using chaotic mapping sequences and then used for the purpose of coefficients in the polynomial function. The permuted secret image is shown in Fig. 6b and resulting image shares are depicted in Fig. 6d. Previously computed set of initial values $\{x'_{01}, x'_{02}, x'_{03}, y'_{01}, y'_{02}, y'_{03}\}$ is used for generating chaotic sequences. The algorithm for image permutation is as follows -

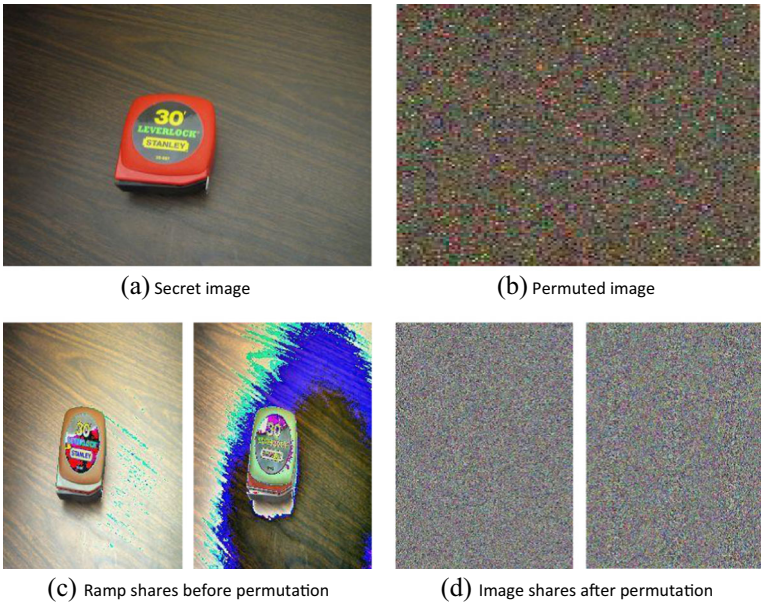


Fig. 6 Different cases of ramp secret sharing

Algorithm_Image permutation

1) Initially, separate RGB color channels $\{S_r, S_g, S_b\}$ of the secret image $S_{p\times q}$ and transform to one-dimensional arrays $r = \{r_1, r_2 \cdots r_{pq}\}$, $g = \{g_1, g_2 \cdots g_{pq}\}$ and $b = \{b_1, b_2 \cdots b_{pq}\}$, where p and q correspond to number of rows and columns of the image.

2) Using x'_{01} as initial value (r), execute chaotic logistic map depicted in (2), in the following loop till the set $r'=\{r'_1, r'_2 \cdots r'_{pq}\}$ of $p \times q$ distinct values is obtained.

Initialize: i=1 and n=0
while(true)
$x_{n+1} = rx_n(1 - x_n)$
$r'_i = \lceil x_i \times pq \rceil$
$n = n + 1, i = i + 1$
end

In the same manner, compute $g'=\{g'_1, g'_2 \cdots g'_{pq}\}$ and $b'=\{b'_1, b'_2 \cdots b'_{pq}\}$ using x'_{02} and x'_{03} as initial values of the chaotic logistic map.

3) Perform pixel permutation on individual color channels $\{S_r, S_g, S_b\}$ of secret image $S_{p\times q}$ according to values of $\{r'_1, r'_2, \cdots r'_{pq}\}$, $\{g'_1, g'_2, \cdots g'_{pq}\}$ and $\{b'_1, b'_2, \cdots b'_{pq}\}$ such that shuffled arrays of pixels for three color channels $P_r = \{p_{1_r}, p_{2_r}, \cdots p_{pq_r}\}$, $P_g = \{p_{1_g}, p_{2_g}, \cdots p_{pq_g}\}$ and $P_b = \{p_{1_b}, p_{2_b}, \cdots p_{pq_b}\}$ are obtained. The scenario is shown in Fig. 7 for RGB channels of S. Permutation operation shuffles pixel positions of $\{r_1, r_2 \cdots r_{pq}\}$, $\{g_1, g_2 \cdots g_{pq}\}$ and $\{b_1, b_2 \cdots b_{pq}\}$ to the positions depict by corresponding arrays $\{r'_1, r'_2 \cdots r'_{pq}\}$, $\{g'_1, g'_2 \cdots g'_{pq}\}$ and $\{b'_1, b'_2 \cdots b'_{pq}\}$.

4) Restate the shuffled pixel values $\{p_{1_r}, p_{2_r} \cdots p_{pq_r}\}$, $\{p_{1_g}, p_{2_g} \cdots p_{pq_g}\}$ and $\{p_{1_b}, p_{2_b} \cdots p_{pq_b}\}$ back to two dimensional matrix $\{P'_r, P'_g, P'_b\}$. Three permuted color channels are obtained as output of this step with random distribution of image pixels forming scrambled images. Figure 8 shows the permuted individual channels of the secret (reference) image obtained as output of this step and it can be clearly observed that no image information is revealed.

5) Similarly, permute pixel values of the target image using $\{y'_{01}, y'_{02}, y'_{03}\}$ as initial values of the chaotic map.

Once permuted, ramp secret sharing is employed to generate image shares. Since RGB color channels of the secret image are already separated and permuted, divide them into non-overlapping blocks such that each block has b number of pixels. The value of b defines degree of the polynomial function depicted in (1), explained more precisely as follows -

$$f(x) = (c_0 + c_1 x + \cdots + c_{b-1}x^{b-1}) mod\ p \tag{16}$$

where $c_0, c_1, \cdots c_{b-1}$ are pixel values of a block. Since 251 is the closest prime number to 255, p can be set as 251. Only pixels values in the range [251–255] need to be truncated to 250. Pixel values of each block are selected and the polynomial function depicted in (16) is employed n times (n is the number of image shares). The reader is advised to [27, 31] for detailed description regarding the value of n. The values obtained from resulting iterations $f(1), f(2), \cdots f(n)$ are sequentially assigned to all the image shares. In this manner, each image share receives only one value corresponding to a particular block and that's why size of each image share is $1/b$ of the secret image. Detailed explanation of share generation is shown in Fig. 9 (a 12×12 image is considered as example) where secret image is divided into non-overlapping blocks $(b_1, b_2, \cdots b_9)$ resulting efficient (size) image shares (of size 3×3). The scenario shown in Fig. 9 is employed to all the color channels

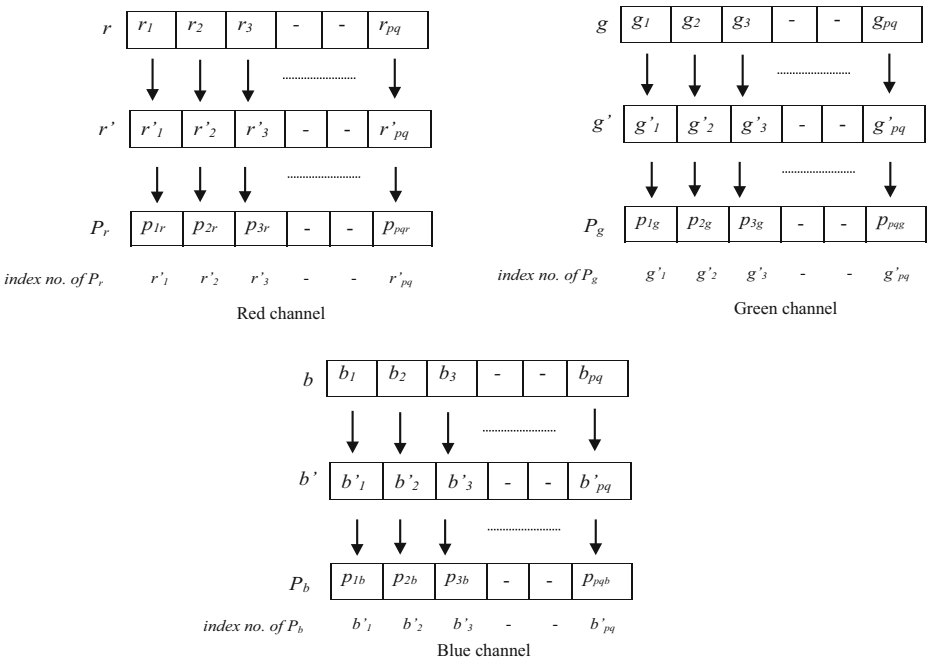


Fig. 7 Image permutation using chaotic sequences

of the secret image. As a result, n image shares are obtained from each color channel. Let $\{f_r(1), f_r(2), \cdots f_r(n)\}$, $\{f_g(1), f_g(2), \cdots f_g(n)\}$ and $\{f_b(1), f_b(2), \cdots f_b(n)\}$ are resulting image shares for red, green and blue color channels, respectively. Sequentially combine

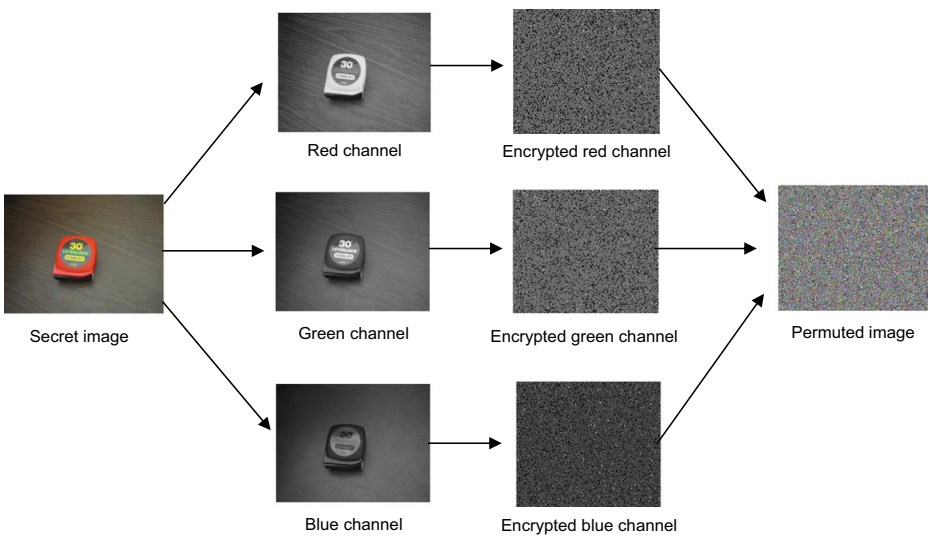


Fig. 8 Encrypted individual channels of the secret image

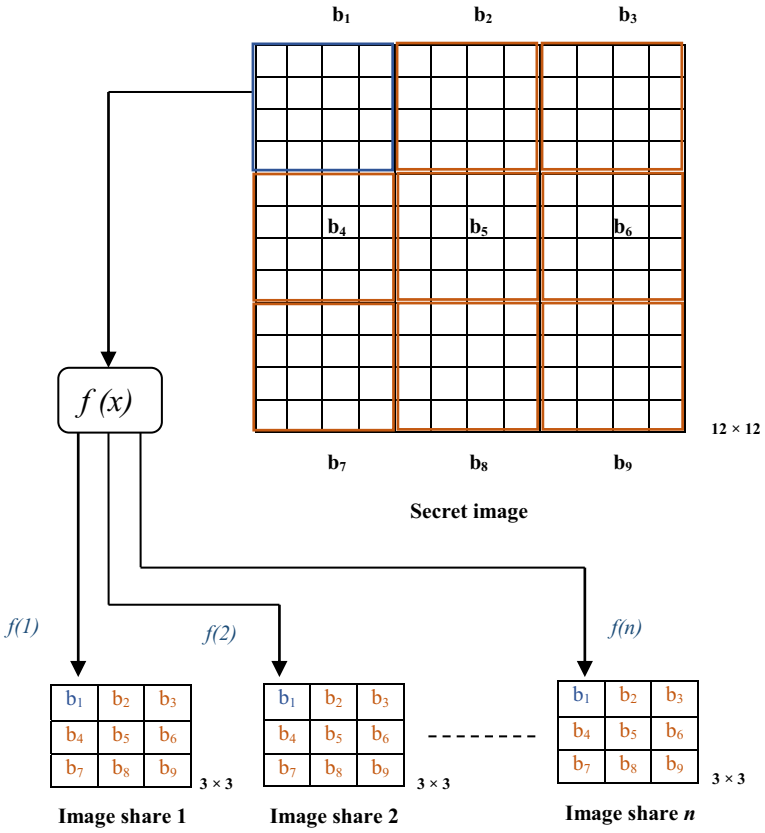


Fig. 9 Description of share generation using ramp secret sharing

each share of red channel with corresponding shares of green and blue color channels to get complete shares, as follows -

$$f_{rgb}(1) = \{f_r(1), f_g(1), f_b(1)\}$$
$$f_{rgb}(2) = \{f_r(2), f_g(2), f_b(2)\}$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$f_{rgb}(n) = \{f_r(n), f_g(n), f_b(n)\}$$

In this manner, complete image shares $\{f_{rgb}(1), f_{rgb}(2), \cdots f_{rgb}(n)\}$ are obtained from individual RGB shares. Since pixel values are directly mapped to coefficients in ramp secret sharing, size of the complete resulting shares is reduced to $1/b$ of the original secret image. The resulting shares are then transmitted to different CDCs for color transfer. Since, image permutation has been accomplished to scramble the pixel values, a single CDC can also be used to store all the image shares. However, we recommend to use multiple CDCs for added security.

3.4 Color transfer at CDC

The pioneering work presented by Reinhard et al. [26] motivated us to employ their method for color transfer over the image shares. Reinhard's method was simple and among the well

known color transfer techniques. They used decorrelated color space for color transfer and believed that the three color channels should be decorrelated such that effects of color transfer may not affect each other's information. Following the same perception, we separated RGB color channels of the secret image while permutation and generation of image shares at the user end. To accomplish the task of color transfer, RGB color channels of the image shares received at CDCs are separately modified and color transfer is performed for individual channels with the help of color transformation values $\{\sigma_r, \sigma_g, \sigma_b\}$. We use Ramp secret sharing to generate multiple image shares. Ramp secret sharing is homomorphic to multiplication. Only the mandatory condition is that each share needs to be processed with the same value. As a consequence, significant results are obtained when statistical transformations are employed for color transfer over all the image shares. On the contrary, reconstruction of the processed image completely fails if less than a threshold number of shares are processed. The algorithm for color transfer in ED is described next and following symbols are used -

E_i: Share i received at CDC$_i$
E_{iRed}: Red color channel of E_i
E_{iGreen}: Green color channel of E_i
E_{iBlue}: Blue color channel of E_i
$\{\sigma_r, \sigma_g, \sigma_b\}$: Set of color transformation values

Algorithm_Encrypted Domain Color Transfer
At each CDC$_i$ do the following -

1) Extract $\forall(E_i)$ $\{E_{iRed}, E_{iGreen}, E_{iBlue}\}$.
2) Compute $\{E'_{iRed}, E'_{iGreen}, E'_{iBlue}\}$ using the set of color transformation values $\{\sigma_r, \sigma_g, \sigma_b\}$ for each pixel at position (i, j) of the image color shares $\{E_{iRed}, E_{iGreen}, E_{iBlue}\}$ as follows -

while(true)

$$E'_{iRed}(i,j) = E_{iRed}(i,j) \times \sigma_r \bmod p$$
$$E'_{iGreen}(i,j) = E_{iGreen}(i,j) \times \sigma_g \bmod p$$
$$E'_{iBlue}(i,j) = E_{iBlue}(i,j) \times \sigma_b \bmod p$$

end

3) Combine the modified shares $\{E'_{iRed}, E'_{iGreen}, E'_{iBlue}\}$ to form a complete processed share E'_i and transmit to the user.

In this manner, all the image shares are processed for color transfer at their respective CDCs and transmitted back to the user. Upon receiving at user end, all the processed image shares are gathered and Lagrange interpolation is employed for polynomial reconstruction. After polynomial reconstruction, extract all coefficients (modified pixel values) and arrange to their respective blocks (as considered during share generation). Now, apply reverse permutation (decryption) over the reconstructed image such that image pixels are rearranged to their original position using the same secret key used while encryption. Furthermore, post-process the decrypted color channels $\{r_d, g_d, b_d\}$ using previously computed set of values $\{r', g', b'\}$ as follows -

$$r^m = r_d - r' \tag{17}$$
$$g^m = g_d - g' \tag{18}$$
$$b^m = b_d - b' \tag{19}$$

Upon successful decryption followed by post-processing mechanisms, it can be found that color palette of the target image is completely transformed as per the colors of the reference image.

3.4.1 Working example

To understand the proposed approach, a working example is demonstrated. For convenience in understanding, we consider only red color channel of the target image and demonstrate the transformation of colors. The matrix we assume is depicted in Fig. 10a and represents the actual pixel values of red color channel of the target image. Since the matrix we assume is very small as compared to the size of digital images and independent of the processing at cloud server, we are omitting the pixel permutation step in the example. The Mean of the red color channel of the target (r_t) image computes to 50.5 and let the same for reference image be 68. Ramp secret sharing is employed with block size (b) = 2 and $p = 251$. Hence, the polynomial function (depicted in (16) for first and second block, B_1 and B_2, computes to -

$$f_{B1}(x) = (49 + 50x) mod\ 251 \tag{20}$$

$$f_{B2}(x) = (51 + 52x) mod\ 251 \tag{21}$$

The computed share values are shown in Fig. 10b. Let the color transformation value σ_r for red channel computes to $\sigma_r = \sigma_{ref}/\sigma_t = \lfloor 1.8991 \rceil = 2$ at the user end. Moreover, another set of value is prepared to post-process the image after receiving from the cloud server as $r' = 50.5 \times 2 - 68 = 33$. The image shares are transmitted to different CDCs along with the color transformation value. As part of the processing, product of color transformation value is accomplished with the image shares at all CDCs and transmitted back to the user end. It is mandatory for all the image shares to be processed using the same color transformation value σ_r failing which, the reconstructed value completely distorts. Upon successful processing at CDCs, the received share values are (198,47) for B_1 and (206,59) for B_2. Since Ramp secret sharing is homomorphic to multiplication, the product of color transformation value and the image shares result to the product of the same value with secret one in the PD

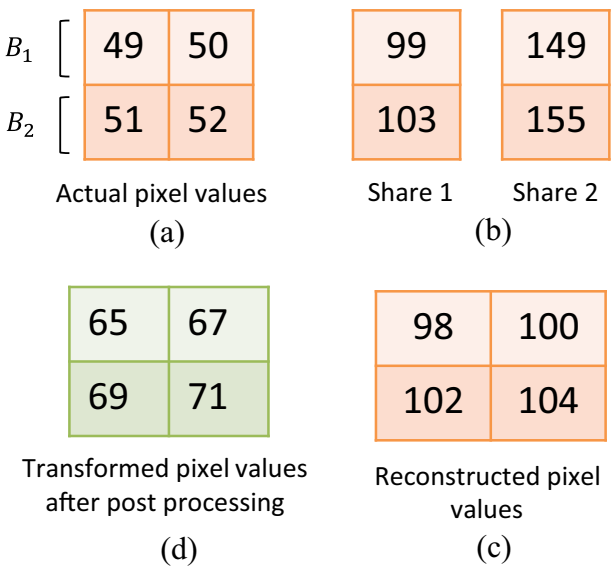


Fig. 10 Working example of the proposed approach

while reconstruction. The reconstructed polynomials are obtained by finding $b-1$ degree Lagrange interpolated polynomial L for B_1 and B_2 as follows -

$$L_{B1}(x) = (98 + 100x) mod\ 251 \tag{22}$$

$$L_{B2}(x) = (102 + 104x) mod\ 251 \tag{23}$$

The coefficients are then extracted from the reconstructed polynomials L_{B1} and L_{B2} and restored to their relative positions forming the processed color matrix, depicted in Fig. 10c. After successful reconstruction at user end, the previously computed value r' is then used to post-process the reconstructed image matrix R' for complete color transformation. Post processing consists of subtraction of r' from R'. The final color transformed matrix is shown in Fig. 10d.

The difference between reconstructed values in PD (depicted in Fig. 1d) and ED (Fig. 10d) is dependent on the fraction of standard deviation scores (σ) of the RGB color channels between reference and target images. For example, suppose the value of σ for red color channel is 2.0001, the rounding function would make it 2 ($\lfloor 2.0001 \rceil = 2$), but the same value is considered if σ is 2.4999. Same problem appears when σ is 2.5001 and rounded to higher value of 3. This situation is considered as the worst case and a maximum loss of 0.4999 would be experienced for σ. A man with bicycles shown in Fig. 14 (second column) is an example of such situation. However, it has been observed that σ is not equal for all three color channels and a balanced color transformation is achieved for various images of datasets. Pleasing colors of the other two images shown in Fig. 14 are the example of such situation.

Uniform colors of the reference and target images are achieved for efficient image stitching using the proposed approach. Figure 11 shows the two cases of image stitching. Image stitching without color transfer is depicted in Fig. 11a and non-uniformity is clearly observed. On the other hand, efficient image stitching is achieved as observed from Fig. 11b where target image is color transformed as per the reference image using the proposed approach.

4 Experimental results

In this section, the proposed approach is assessed for color transfer operations in encrypted domain. Since the proposed approach is among the first ventures to perform ED color transfer, results are compared with the existing methods of PD.

4.1 Test images

To test the effects of color transfer, we tested our approach with the following three datasets -

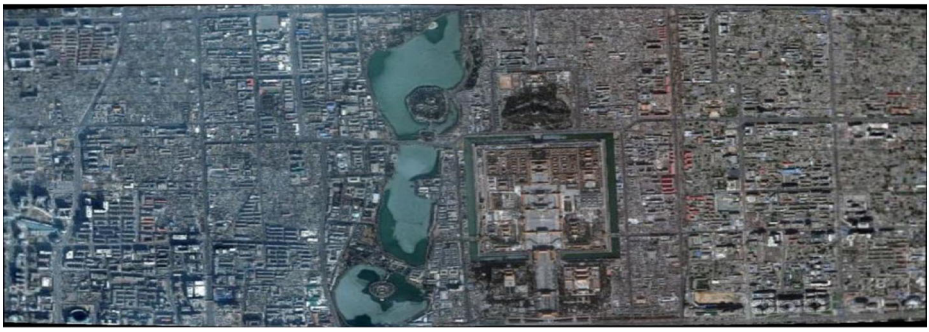
1) Berkeley BSDS500 image segmentation dataset [19]. The dataset consists of total 500 images. Images are categorized into three subsets; *Test*, *Train* and *Val* each with 200, 200 and 100 images, respectively.

2) INRIA Person Dataset [3]. 210 images of people and bikes at various outdoor locations are present under different lighting/expressions/backgrounds.

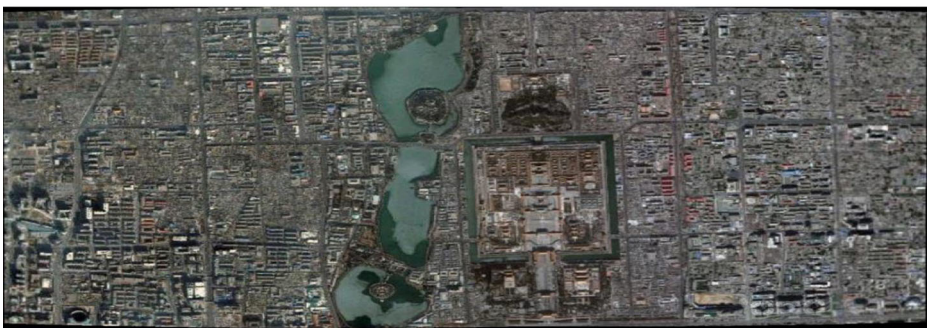
3) IITR-Tennis Court Dataset [2]. We have created this dataset to test the effects of single dominating color tone over color transformations. The dataset consists of total six real and synthetic image pairs for the purpose of color transfer and is available at [2].

Reference image

Target image



(a) Image stitching without color transfer



(b) Image stitching with color transfer

Fig. 11 Different cases of image stitching

4.2 Performance metric

We have used color similarity [36] for assessment of color transfer operations between reference and reconstructed images. Color similarity $CS(ref, r)$ among reference and reconstructed image ref and r, is evaluated as follows -

$$CS(ref, r) = PSNR(ref', r') \tag{24}$$

where, peak signal to noise ratio is designated by $PSNR = 20 \times log_{10}(G/RMSD)$. G is largest feasible pixel value of the image and RMSD is root mean square difference between ref and r. The overlapped region among ref and r is represented by ref' and r' respectively. According to Xu et al. [36], higher score of CS belongs to more color similarity between corrected and reference images.

4.3 Color transfer over single dominating color tone

As discussed in Section 4.1, we use IITR-Tennis Court dataset to test the effects of color transfer over single dominating color tone. The dataset consists of six real images and each one is set to the intense of one color balance axis, forming corresponding synthetic images. The real images available in the dataset are shown in Fig. 12a and their corresponding synthetic images each with a dominating color tone are depicted in Fig. 12b. These images

(a) Original images (b) Synthetic images (c) Proposed approach (d) Reinhard et al. [26]

Fig. 12 Results of different images with single dominating color tone (best viewed in color display)

are influenced by red, green, blue, yellow, magenta and cyan colors respectively. Original images shown in Fig. 12a are considered as reference images whereas the images shown in Fig. 12b as target images. Privacy preserving color transfer between these image pairs is performed using the proposed approach and resulting images are depicted in Fig. 12c. For comparison purpose, Reinhard's scheme [26] (PD) is employed over the same image pairs and resulting images are shown in Fig. 12d. As observed from Fig. 12c, visual quality of the resulting images using our approach in ED is highly acceptable, and equivalent to the images obtained in PD using original Reinhard's scheme ($l\alpha\beta$ color space). Moreover, we have computed CS scores between single color dominating image pairs shown in Fig. 12a and b, and results are shown in Table 1. A *virtual baseline approach* is used for the purpose of evaluation. It resembles *no transfer*, meaning that color similarity scores are computed directly between reference and target images without any color transfer. The images from top (towards bottom) are considered and named as $a_1, b_1, c_1, d_1, e_1, f_1$ in Table 1. As observed, the proposed approach achieves significant gain % over the baseline approach for image pairs with single dominating color tones. The gain % is computed as -

$$gain\% = [1 - (CS_{baseline}/CS_{proposed})] \times 100 \tag{25}$$

where $CS_{baseline}$ and $CS_{proposed}$ are CS scores of the baseline and proposed approach, respectively.

4.4 Experiments using image processing tool

The proposed approach is employed for color transfer in ED and results are shown in Figs. 13 and 14 for sample images of BSDS500 and INRIA Person datasets respectively. In the same manner as of [36], each image pair is generated using following three steps: First, a set of poor exposure level images from the dataset has been prepared. Then, color levels of the selected images are auto adjusted using an image processing tool [1], for new images of the same scene, but with different color properties. In the end, a visual assessment among new and original image is performed and the image having better quality among the two is selected as reference image and other as target image. Random assignment is made if quality of both the images is similar. Afterwards, the proposed scheme is employed with newly formed image pairs and its caliber to enhance color palette of the target image by modifying its color distribution in ED is assessed.

The mean CS scores computed for all images of the selected datasets using the proposed approach and comparison with baseline as well as Reinhard's scheme ($l\alpha\beta$ color space) are shown in Table 2. It can be observed from Table 2 that CS scores obtained for images processed with the proposed approach are almost near to the Reinhard's scheme. Additionally,

Table 1 CS for images of IITR-Tennis Court dataset with single dominating color tone

Image name	Reinhard et al. in PD [26]	Proposed approach in ED	Baseline approach in PD	Gain %
a_1	12.5084	12.0159	2.4026	80.0048%
b_1	6.8988	6.5994	1.3637	79.3360%
c_1	3.4383	6.4278	2.9085	54.7512%
d_1	12.2132	11.1523	6.1262	45.0678%
e_1	6.9606	6.1067	5.1385	15.8547%
f_1	6.7368	6.1864	3.9797	35.6701%

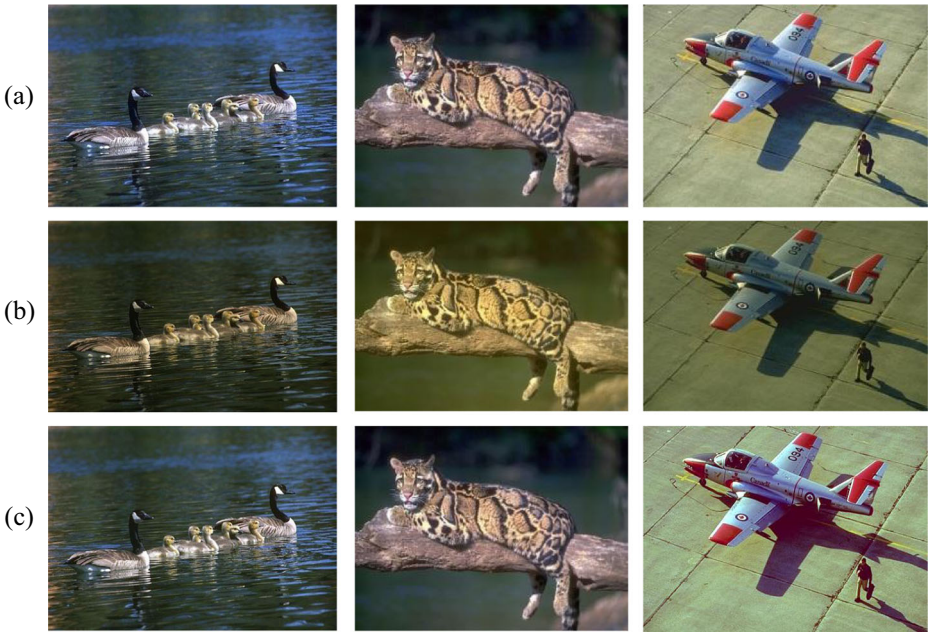


Fig. 13 Sample images of BSDS500 image set: **a** reference image; **b** target image; **c** color transformed image using the proposed approach

Fig. 14 Sample images of INRIA Person image set: **a** reference image; **b** target image; **c** color transformed image using the proposed approach

Table 2 Mean CS for images of various datasets

Image set	Reinhard et al. in PD [26]	Proposed approach in ED	Baseline approach in ED	Gain %
Test	11.4268	9.6811	3.3764	65.1238%
Train	10.7372	8.2846	1.3457	83.7566%
Val	10.0820	9.8953	1.1565	88.3126%
INRIA	24.92767	24.6040	11.65669	52.6231%

gain % of the CS scores obtained by the proposed approach over the baseline is superior, employing that quality of the color transfer is highly effective. As observed from Tables 1 and 2, following conclusions are drawn -

1. The CS scores are evaluated for all image pairs of the selected datasets and higher CS scores are obtained as compared to the baseline approach.
2. It can be observed from Figs. 13 and 14 that visual effects of performing color transfer in ED are superior and acceptable.
3. The proposed approach uses permutation and modifies pixels sequence spatially. However, color information still holds which is further modified using ramp secret sharing. Security is ensured by the shares generated from the polynomial function as well as secret key for chaotic sequences (as any small change in secret key may result completely different chaotic sequence making it impossible to decrypt).
4. Primary objective of Reinhard's scheme [26] is to efficiently transfer colors of target image as per the reference image and hence decorrelated color space is used for the purpose. On the other hand, our proposed scheme separates the RGB color channels while generating image shares and efficient color transfer is achieved making the separation significantly effective as of the original Reinhard's scheme.

5 Security analysis

The proposed approach is tested for all possible attacks and results are discussed in this section. Since security of the existing image encryption schemes is analyzed using the standard gray-scale *Lena* image, we have also performed all the tests over standard gray-scale *Lena* image (shown in Fig. 15a) of size 512×512 such that constructive comparison between existing image encryption schemes and the proposed approach can be established.

5.1 Robustness of ramp secret sharing scheme

Theorem 1 *Any* $b - 1$ *or less shares are insufficient to reveal the secret image.*

Proof Let size of the secret image is $m \times n$. In order to employ ramp secret sharing, secret image is divided into non-overlapping blocks with b number of pixels in each block (total number of blocks = $m \times n/b$). As discussed in Section 3.3, ramp secret sharing requires all pixel values of a block as coefficients in the polynomial function (depicted in (16). As a consequence, total $m \times n/b$ polynomials are required (one polynomial per block). Hence, in order to reconstruct b coefficients $(c_0, c_1 \cdots c_{b-1})$ of a single polynomial, we

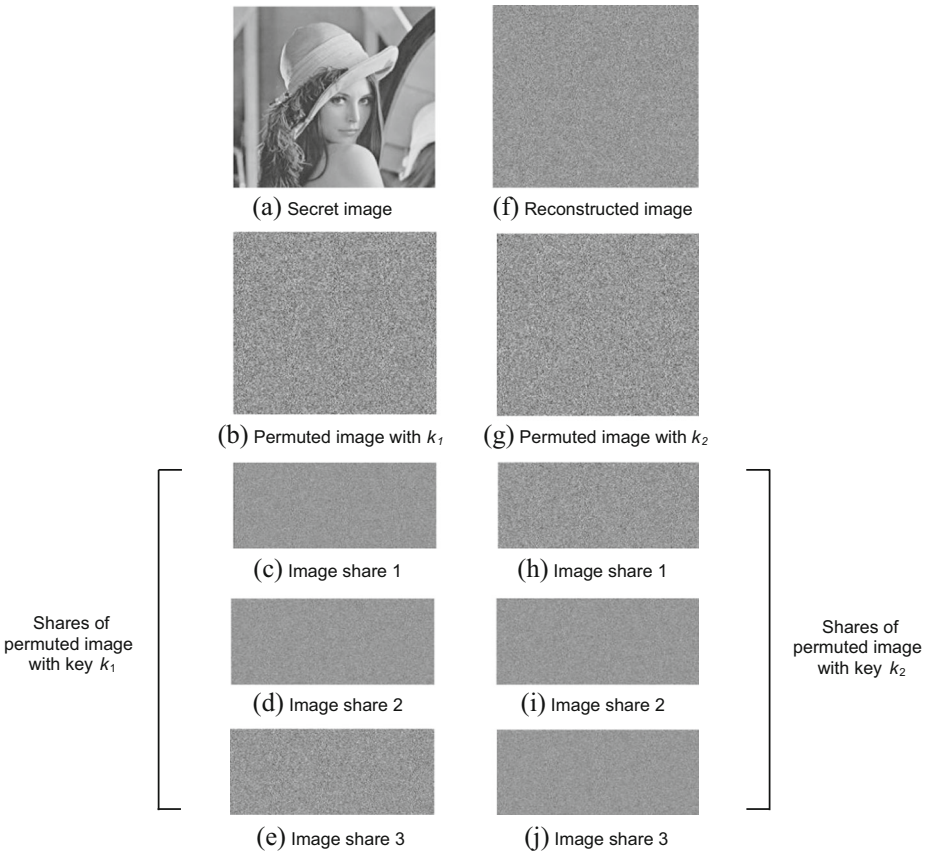


Fig. 15 Key sensitivity analysis

need b equations. If less than b image shares $(b-1)$ are received, only $b-1$ equations are constructed from $f'(1), f'(2) \cdots f'(b-1)$. Since $f'(1) = (c_0 + c_1 + \cdots + c_{b-1}) mod\ p$, $f'(2) = (c_0 + 2c_1 + \cdots + 2^{b-1}c_{b-1}) mod\ p \cdots f'(b-1) = (c_0 + (b-1)c_1 + \cdots + (b-1)^{b-1}c_{b-1}) mod\ p$, solving these $b-1$ equations for b unknowns leads to 251 possible solution sets. The first set belongs to $c_0 = 0$, second belongs to $c_0 = 1$, and moving sequentially, the 251^{st} set belongs to $c_0 = 250$ (as all pixel values are truncated in the range of 0-250). Hence, the possibility of identifying the ideal solution is only 1/251. This is the case when only one pixel is considered. Moreover, for complete image (with $m \times n/b$ blocks), it becomes

$$(1/251)^{m \times n/b} \tag{26}$$

which is sufficient to resist the adversary from obtaining any meaningful information from the image shares. Additionally, secret key based image permutation is performed before employing ramp secret sharing which acts as added advantage (for securing the image shares). □

5.2 Key space analysis

Key space of a cryptosystem is supposed to be fairly large to refuse Brute-force attacks. If key space is small, the possibilities are high for Brute-force attacks to be

possible. Total key length of 360-bits is employed to compute different initial values $\{x'_{01}, x'_{02}, x'_{03}, y'_{01}, y'_{02}, y'_{03}\}$ of one dimensional chaotic map at different phases of the proposed approach. Such large key space of 360-bits causes 2^{360} different combinations which is ample to resist various brute-force attacks.

5.3 Key sensitivity

Key sensitivity is one of the important aspects to be considered while designing a good image cryptosystem. Any small modification in the secret key should result an absolutely different image. The proposed approach possesses good key sensitivity properties as analyzed by reconstructing the same image with a single bit modification in the secret key. As a result, actual secret image is not recovered with such modification. Figure 15b and g shows permuted images with two different keys k_1 and k_2 (with a single bit difference). Figure 15c–e and h–j shows the image shares generated from the permuted images shown in Fig. 15b and g respectively. While reconstructing from image shares of the permuted image using key k_1 and further decrypting using key k_2 (or vice-versa), we found that reconstructed image is completely distorted and no image information is revealed. The reconstructed image is shown in Fig. 15b.

5.4 Histogram analysis

A histogram represents the graphical distribution of image pixels. Since pixel intensity values are modified while encryption, histogram of the encrypted image should be modified in a manner such that no image information can be obtained. To test the histogram of encrypted images, we consider the *Lena* image, shown in Fig. 16a and employed the proposed encryption algorithm (for two shares). The image shares are shown in Fig. 16b and c. Figure 16e and f shows that the histograms of image shares are completely different from the histogram

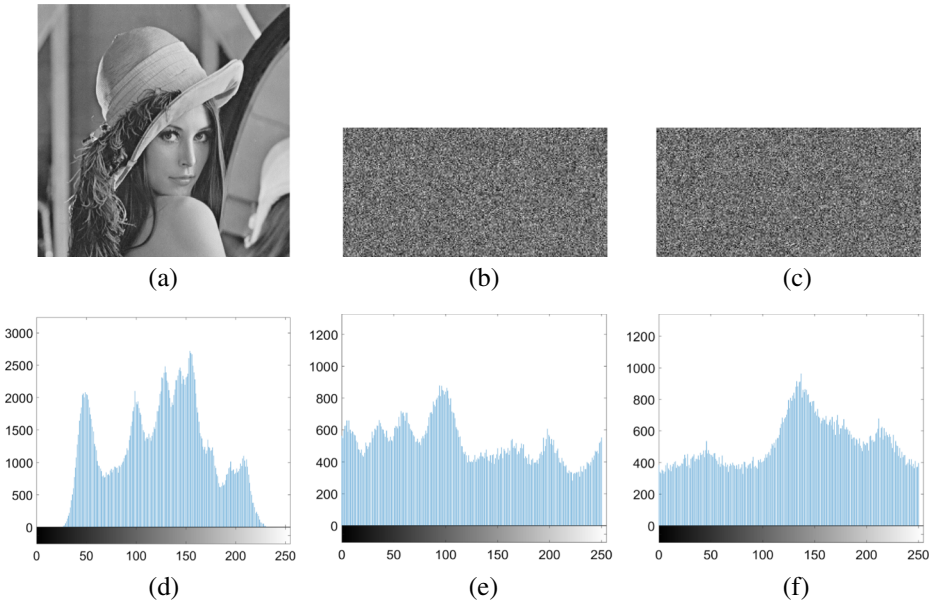


Fig. 16 Histogram analysis

of the plain secret image, shown in Fig. 16d. Hence, no secret information is revealed from histogram of the image shares using the proposed approach.

5.5 Correlation coefficient of adjacent pixels

An effective image cipher should be robust against various statistical attacks. Correlation between adjacent pixels in plain image is very high and, on the contrary, should be very low for the encrypted image. The correspondence between adjacent pixels of the encrypted image should be distorted such that image information cannot be obtained from relative values of the pixels. In order to test the correlation between adjacent pixels of the image shares, we consider horizontal, vertical as well as diagonal directions. The encryption scenario is executed multiple times and average values of the correlation test results for *Lena* image (shown in Fig. 15a) are computed. Table 3 shows comparison of correlation coefficients between the proposed approach and existing image encryption schemes [10, 22, 30, 33] in the literature. Since multiple image shares are generated using the proposed approach, we have computed average values of the image shares and used in Table 3 for the comparison purpose. While the proposed approach provides the facility of performing color transfer between encrypted images, still superior correlation test results are obtained as compared to existing image encryption schemes. The correlation coefficients are calculated using following equations -

$$Mean(p) = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{27}$$

$$Variance(x) = \frac{1}{N}\sum_{i=1}^{N}(p_i - Mean(p))^2 \tag{28}$$

$$Covariance(p,q) = \frac{1}{N}\sum_{i=1}^{N}(p_i - Mean(p))(q_i - Mean(q)) \tag{29}$$

$$r_{pq} = \frac{Covariance(p,q)}{\sqrt{Variance(p)} \times \sqrt{Variance(q)}} \tag{30}$$

where, p and q are values of two neighboring pixels of the image.

5.6 Differential analysis

Differential analysis is performed to test the relation between plain and encrypted images. In order to perform differential attacks, the adversary tries to employ a slight change in the plain secret image such that any information in relation to the plain and encrypted images can be obtained. Hence, two measures; *Number of Pixels Change Rate* (NPCR) and *Unified Average Changing Intensity* (UACI) are employed to test effectiveness of the proposed

Table 3 Comparison of correlation coefficients

Neighboring direction	Plain Image *Lena*	Proposed approach	Teng et al. [30]	Wang et al. [33]	Chong et al. [10]	Pareek et al. [22]
Horizontal	0.9404	−0.0067	0.024178	0.0321067	0.0088	0.0031
Vertical	0.9299	−0.0012	−0.019425	0.0271879	−0.0087	−0.0016
Diagonal	0.9257	0.0059	0.024322	0.0383929	−0.0060	0.0067

approach against differential attacks. NPCR is used to identify the percentage of different pixels between resulting encrypted images when a pixel value of the plain secret image is slightly modified. On the other hand, UACI is used to obtain average intensity differences between two encrypted images. The NPCR and UACI are evaluated by the following equations -

$$NPCR = \frac{\sum_{ij} D(i, j)}{M \times N} \times 100 \tag{31}$$

$$UACI = \frac{1}{M \times N} \left[\sum_{ij} \frac{E_1(i, j) - E_2(i, j)}{255} \right] \tag{32}$$

where, E_1 and E_2 are encrypted images computed with only one pixel value difference between their corresponding original images. M and N are width and height of the image. $D(i, j)$ is a two-dimensional array derived from E_1 and E_2 in the following manner -

$$\begin{aligned} &\textit{if } E_1 = E_2 \\ &\quad D(i, j) = 0 \\ &\textit{else} \\ &\quad D(i, j) = 1 \end{aligned}$$

To test the proposed approach against differential attacks, we have generated two different permuted images with respect to two plain images having slight difference in only one pixel value (secret key is same in both the scenarios). Furthermore, image shares are generated from the two permuted images and average of NPCR and UACI test results between the corresponding image shares is shown in Table 4. Comparison between the proposed approach and existing image encryption schemes [10, 22, 30, 33] is also depicted and it can be observed from Table 4 that the proposed approach exhibit good NPCR and UACI scores as compared to the existing image encryption schemes and suffice to resist the differential attacks.

5.7 Analysis of cloud computing security threats

Various threats exist in cloud computing environments. Data sharing among multiple users and open access over cloud services makes these infrastructures vulnerable to deploy efficiently. Cloud Security Alliance (CSA) released security guidelines for managing risks and understanding security threats in cloud computing [28]. Considering the various threats released by CSA, our proposed approach ensures security of the secret images under following threat categories [8].

Account or service hijacking The attackers can steal and access confidential information from critical areas of cloud computing. Our proposed approach makes the attackers unable to steal and access confidential information as images are stored and processed in encrypted form over cloud.

Table 4 NPCR and UACI comparison

NPCR/ UACI	Proposed scheme	Teng et al. [30]	Wang et al. [33]	Chong et al. [10]	Pareek et al. [22]
NPCR	99.6501	93.6768	99.6170	99.6100	96
UACI	33.5691	33. 3364	33.4933	≈ 33	31.79

Insecure interface Cloud computing customers use Application Programming Interface (API) or other software interface to deal with cloud services. Deploying cloud services with our proposed approach enforce secret images to pass through such API's. However, no image information is revealed at any intermediate level.

Malicious insider Malicious insider is organization's own employee acting as adversary. Such dangerous threat affects the confidentiality, integrity and availability of the organization's data. While multiple image shares are generated (from the permuted image) and distributed to different CDCs using the proposed approach, any malicious insider cannot access full contents of the secret image.

Data breaches The problem of illegal viewing of data by competitors can be resolved by processing the secret images in encrypted form. As a consequence, neither cloud owner nor any competitor can illegally view the image information.

Insecure VM migration Attackers can access data while moving VMs (Virtual Machines) and even sometimes transfer the complete VM to another host. In such scenario, if the VM is moved to another untrusted host, the proposed approach assures privacy of image information.

6 Conclusion

An efficient approach for privacy preserving color transfer and storage over third party cloud infrastructures is proposed in this paper. Secret images are encrypted and processed for color transfer without revealing any image information at the cloud data centers. Additionally, the proposed approach enables cloud infrastructures to facilitate the use of huge storage pools in a privacy preserving manner too. Comparison with existing schemes has been done for verification as well as security analysis, and effective results are achieved. Since we are among the initial ventures to perform the task of color transfer in ED, we believe that our approach is a significant initiative towards color transfer when the test images are in encrypted form. Future work would be fixing the occurrence of color transformation errors due to rounding function for more accurate results as of the PD.

Acknowledgements This work was supported by Information Security Education and Awareness (ISEA) Project (phase II), Deity, Government of INDIA.

Compliance with Ethical Standards

Conflict of interests All authors declare that they have no conflicts of interest regarding the publication of this manuscript.

References

1. I. Skiljan.irfanview http://www.irfanview.com. Accessed: 2017-04-26
2. Iitr-tennis court dataset https://sites.google.com/site/amiteshrajput/. Accessed: 2017-05-09
3. Inria person dataset http://pascal.inrialpes.fr/data/human/. Accessed: 2017-05-11

4. Al-Otaibi NA, Gutub AA (2014) 2-leyer security system for hiding sensitive text data on personal computers. Lect NoteS Inform Theory 2:151–157
5. Al-Otaibi NA, Gutub AA (2014) Flexible stego-system for hiding text in images of personal computers based on user security priority. In: International conference on advanced engineering technologies (AET-2014), pp 250–256
6. Alassaf N, Alkazemi B, Gutub A (2003) Applicable light-weight cryptography to secure medical data in iot systems. Arabia
7. Alotaibi N, Gutub A, Khan E (2015) Stego-system for hiding text in images of personal computers. In: The 12th learning and technology conference: Wearable tech/wearable learning
8. Amini A, Jamil N, Ahmad A, Z'aba M (2015) Threat modeling approaches for securing cloud computing. J Appl Sci 15(7):953
9. Finlayson GD, Mackiewicz M, Hurlbert A (2015) Color correction using root-polynomial regression. IEEE Trans Image Process 24(5):1460–1470
10. Fu C, Chen JJ, Zou H, Meng WH, Zhan YF, Yu YW (2012) A chaos-based digital image encryption scheme with an improved diffusion strategy. Optics Express 20(3):2363–2378
11. Gutub AAA et al (2010) Pixel indicator technique for rgb image steganography. Journal of Emerging Technologies in Web Intelligence 2(1):56–64
12. Gutub AAA, Khan FAA (2012) Hybrid crypto hardware utilizing symmetric-key and public-key cryptosystems. In: International conference on advanced computer science applications and technologies (ACSAT), pp 116–121. IEEE
13. Gutub A, Ankeer M, Abu-Ghalioun M, Shaheen A, Alvi A (2008) Pixel indicator high capacity technique for rgb image based steganography. In: International conference on advanced computer science applications and technologies (ACSAT)
14. Hsu CY, Lu CS, Pei SC (2012) Image feature extraction in encrypted domain with privacy-preserving sift. IEEE Trans Image Process 21(11):4593–4607
15. Khan F, Gutub AAA (2007) Message concealment techniques using image based steganography. IEEEGCC 2007
16. Lathey A, Atrey PK (2015) Image enhancement in encrypted domain over cloud. ACM Trans Multimed Comput Commun Appl (TOMM) 11(3):38
17. Lu H, Li Y, Serikawa S (2013) Underwater image enhancement using guided trigonometric bilateral filter and fast automatic color correction. In: 20Th IEEE international conference on image processing (ICIP), pp 3412–3416. IEEE
18. Ly DS, Beucher S, Bilodeau M (2014) Color correction through region matching leveraged by point correspondences. In: IEEE International conference on image processing (ICIP), pp 640–644. IEEE
19. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: 8th IEEE international conference on computer vision, 2001 (ICCV), vol 2, pp 416–423. IEEE
20. Mohanty M, Ooi WT, Atrey PK (2013) Scale me, crop me, knowme not: Supporting scaling and cropping in secret image sharing. In: IEEE international conference on multimedia and expo (ICME), pp 1–6. IEEE
21. Nanda H, Cutler R (2001) Practical calibrations for a real-time digital omnidirectional camera. CVPR Technical Sketch 20:1–4. IEEE
22. Pareek NK, Patidar V, Sud KK (2013) Diffusion–substitution based gray image encryption scheme. Digital signal processing 23(3):894–901
23. Pitié F, Kokaram AC, Dahyot R (2007) Automated colour grading using colour distribution transfer. Comput Vis Image Underst 107(1):123–137
24. Rahulamathavan Y, Phan RCW, Chambers JA, Parish DJ (2013) Facial expression recognition in the encrypted domain based on local fisher discriminant analysis. IEEE Trans Affect Comput 4(1):83–92
25. Rajput AS, Raman B (2017) Color me, store me, know me not: Supporting image color transfer and storage in encrypted domain over cloud. In: IEEE International conference on multimedia & expo workshops (ICMEW), pp 291–296. IEEE
26. Reinhard E, Adhikhmin M, Gooch B, Shirley P (2001) Color transfer between images. IEEE Comput Graph Appl 21(5):34–41
27. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
28. Soares LF, Fernandes DA, Freire MM, Inácio PR (2013) Secure user authentication in cloud computing management interfaces. In: 32nd IEEE international performance computing and communications conference (IPCCC), pp 1–2. IEEE

29. Tai YW, Jia J, Tang CK (2005) Local color transfer via probabilistic segmentation by expectation-maximization. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 1, pp 747–754. IEEE
30. Teng L, Wang X (2012) A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive. Opt Commun 285(20):4048–4054
31. Thien CC, Lin JC (2002) Secret image sharing. Comput Graph 26(5):765–770
32. Uyttendaele M, Eden A, Skeliski R (2001) Eliminating ghosting and exposure artifacts in image mosaics. In: IEEE Computer society conference on computer vision and pattern recognition (CVPR), vol 2, pp II–II. IEEE
33. Wang X, Jin C (2012) Image encryption using game of life permutation and pwlcm chaotic system. Opt Commun 285(4):412–417
34. Wu Y, Yang G, Jin H, Noonan JP (2012) Image encryption using the two-dimensional logistic chaotic map. Journal of Electronic Imaging 21(1):013,014–1
35. Xiang Y, Zou B, Li H (2009) Selective color transfer with multi-source images. Pattern Recogn Lett 30(7):682–689
36. Xu W, Mulligan J (2010) Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 263–270. IEEE
37. Yan WQ, Kankanhalli MS (2015) Face search in encrypted domain. In: Pacific-rim symposium on image and video technology, pp 775–790. Springer
38. Zafar F, Khan A, Malik SUR, Ahmed M, Anjum A, Khan MI, Javed N, Alam M, Jamil F (2017) A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends. Computers & Security 65:29–49

Amitesh Singh Rajput received Bachelor of Engineering degree from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal and Master of Technology from School of Information Technology, University Teaching Department, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal. He is currently doing Ph.D. from Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. His area of research include Image Processing, Encrypted Domain Processing and Cloud Computing.

Balasubramanian Raman Associate Professor in the Department of Computer Science and Engineering at Indian Institute of Technology Roorkee, obtained his B.Sc Degree in Mathematics from A.M. Jain College (University of Madras) in 1994, M.Sc degree in Mathematics from Madras Christian College (University of Madras) in 1996 and Ph.D from Indian Institute of Technology Madras in 2001. He was a Post Doctoral Fellow at University of Missouri Columbia, USA in 2001-02 and a Post Doctoral Associate at Rutgers, the State University of New Jersey, USA in 2002-03. He joined Department of Mathematics at Indian Institute of Technology Roorkee as Lecturer in 2004 and became Assistant Professor in 2006. He was a Visiting Professor and a member of Computer Vision and Sensing Systems Laboratory in the Department of Electrical and Computer Engineering at University of Windsor, CANADA during May - August 2009. His area of Research includes Vision Geometry, Digital Watermarking using Mathematical Transformations, Image Fusion, Biometrics, Secure Image Transmission over Wireless Channel, Content Based Image Retrieval and Hyperspectral Imaging. He has more than 150 research publications in reputed journals and conference proceedings.