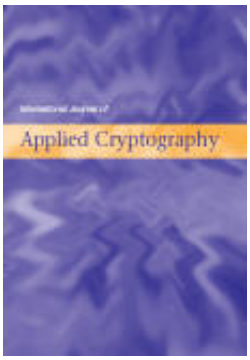


International Journal of Applied Cryptography

ISSN online: 1753-0571 - ISSN print: 1753-0563
https://www.inderscience.com/ijact

Image and object encryption using multiplicative cipher and K-means clustering algorithm

Maroti Deshmukh, Arjun Singh Rawat

DOI: 10.1504/IJACT.2023.10060005

Article History:

Received:	14 January 2023
Last revised:	03 April 2023
Accepted:	01 June 2023
Published online:	03 May 2024

Image and object encryption using multiplicative cipher and K-means clustering algorithm

Maroti Deshmukh* and Arjun Singh Rawat

Department of Computer Science and Engineering,
National Institute of Technology,
Uttarakhand, India
Email: marotideshmukh@nituk.ac.in
Email: arjunsinghrawat005@gmail.com
*Corresponding author

Abstract: In recent years, the development of various visual sensing and image analysis techniques has resulted in the creation of images that contain extremely sensitive data. Unauthorised individuals who access this data illegally risk capturing and disclosing all the sensitive information. To address this issue, we propose a simple and effective image and object encryption approach using a multiplicative cipher and K-means clustering algorithm. The proposed approach involves two levels of encryption, object detection, and K-means clustering in two different phases. In phase 1, the main object from the original image is encrypted using a multiplicative cipher. Phase 2 uses the K-means clustering technique to encrypt the noisy image generated in phase 1. The decryption process is similar to the encryption process but is carried out in reverse order. Moreover, the proposed approach is indeed lossless, even if data is encrypted multiple times. Furthermore, the proposed technique is demonstrated to be robust to differential attacks and resistant to statistical attacks. The results of different experiments show that the approach is effective, secure, and suitable for a wide range of industrial applications.

Keywords: object detection; K-means clustering; edge detection; image encryption; object encryption; multiplicative cipher; decryption.

Reference to this paper should be made as follows: Deshmukh, M. and Rawat, A.S. (2023) 'Image and object encryption using multiplicative cipher and K-means clustering algorithm', *Int. J. Applied Cryptography*, Vol. 4, Nos. 3/4, pp.195–204.

Biographical notes: Maroti Deshmukh is an Assistant Professor in Computer Science and Engineering at the National Institute of Technology, Uttarakhand. He received his PhD from the MNIT Jaipur, MTech from the Hyderabad Central University and BTech from the SGGSIE&T Government Engineering College Maharashtra. His research area includes secret sharing schemes, cryptography, and machine learning. Currently, he is working in the area of deep neural networks and machine learning. He published his work in more than 23 reputed international journals and 22 international conferences. He has more than nine years of teaching and research experience.

Arjun Singh Rawat is currently pursuing his PhD in Computer Science and Engineering at the National Institute of Technology, Uttarakhand. His area of interest is in cryptography, secret sharing schemes, machine learning, deep neural networks, medical data analysis, and blockchain. He has technical skills in programming languages like MATLAB, C, C++, Java, Python, and PHP. He has published more than 14 publications in reputed international journals and conferences.

1 Introduction

The internet is a global medium that is used to transfer data from one place to another. Data can be transmitted via text, email, audio, video, or other formats. Images and videos are used often today, as seen by platforms like Facebook, Snapchat, Instagram, WhatsApp, and others. However, there are certain issues with data transmission, the major issues include security and authenticity. Security is the defence of digital information and IT assets against internal and external, malicious and accidental threats (Stallings, 2006). Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. Security is the most crucial need that is required in the modern world. As the internet is growing and so is its usage there is a need for hiding the content of files that are being transferred from one node of the internet to another, and for that purpose, people use different cryptographic algorithms (Rawat and Deshmukh, 2019, 2020a, 2020b, 2021a, 2021b).

The algorithms that are used for encryption of files are simple like additive cipher, and multiplicative cipher and some are very complex, e.g., AES, DES, etc. Most of these algorithms use a key pair mechanism in which the encrypted file will only be decrypted when the right key is used. Image cryptography (Forouzan and Mukhopadhyay, 2011) is a special encryption technique to hide information in images in such a way that it can be decrypted by the human vision if the correct key image is used. This technique can be used to protect images on storage devices such as computers or hard disks, etc. It can also be used to protect the image in transmission over network(s). Image security has become an important concern because of so many web attacks that are possible with the advancements in the technology. The government, military, and hospitals deal with confidential images about their financial status, enemy positions, patients respectively. Most of this information is transmitted over the internet. If the confidentiality of these images were broken then it may lead to economic crisis, declaration of war, wrong treatment, etc. Protecting the images is a valid and an important requirement.

Many techniques based on image encryption (Ben Slimane et al., 2018; Alexan et al., 2021; Elkandoz and Alexan, 2022; Luo et al., 2019; Arab et al., 2019; Wang et al., 2019; Lu et al., 2020; Man et al., 2021; Gupta et al., 2022) have been proposed for exchanging the multimedia data as image securely over a public channel with less computation cost. The existing schemes deal with complete-image encryption. In any case, if an attack is made on an image that reveals a portion of an object, that tiny portion of the object may hint at the actual object, which could be problematic for identifying the original information. Therefore, to overcome this issue we propose a new approach, the approach first encrypts the object of an image then performs complete image encryption, which makes it more challenging throughout the attack than the existing technique to divulge the object details.

The proposed approach is based on image and object encryption using a multiplicative cipher and K-means clustering algorithm. This approach firstly tries to find out the major object(s) in the image and then apply multiplicative cipher (Kahate, 2013) to encrypt the object only. Object detection plays an important role in this model. As using image encryption all we want is to hide the details of the object or the secret data that is in the image so the best way is to first encrypt the object and then encrypt the complete image. It will be difficult for attacker to get full information even it decrypts the first layer. Key chosen for the multiplicative cipher is found using edge detection (Jain, 1989) algorithm. When the object has been encrypted we then encrypt the whole image using multiplicative cipher whose key will again be found out using edge detection algorithm. As the range of modulo increases in the multiplicative cipher the encryption becomes harder to decrypt and in images there are 255 possible values for a intensity so it will become harder for the hacker to guess values for such larger number of pixels. In multiplicative cipher, a number is selected whose inverse exist in the modulo (n). That number will then be used to encrypt the data and its inverse is used to decrypt the data. Any value other than the inverse cannot decrypt the data. After the complete image has been encrypted, we then divide the image into clusters and then apply multiplicative cipher individually on each cluster using the cluster indices (Burkardt, 2009) and cluster centroids that are returned when we apply the K-means clustering. K-means is a unsupervised learning algorithm used to solve the problem of clustering. In this algorithm, the given data is classified into different clusters that have certain properties in common. All these clusters have a centroid around which the data is clustered into different clusters. The centroids are changed with every iteration and comes in steady state after some iterations and after that solution is obtained. The decryption works in the similar fashion but in reverse order and we are able to get the original image with no loss of information. The images used in this algorithm are confined to be only greyscale images.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{m} ||x_i^j - c_j||^2 \tag{1}$$

where $||x_i^j - c_j||^2$ is a chosen distance measure between a data point x_i^j and the cluster centre, d is an indicator of the distance of the n data points from their respective cluster centres.

In this model, after encrypting the object in the image we have used to K-means clustering algorithm to divide the encrypted image into a number of clusters so as different cluster can be encrypted with different key so that even if attacker is able to identify any one of the key he/she may not be able to retrieve the complete image. Our approach also provides additional security benefits as it prevents potential adversaries from obtaining information about the encrypted objects, which could be particularly useful in situations where only specific objects need to be protected. This is because our approach considers the object as the most relevant aspect of the image and therefore first encrypts the object before encrypting the complete image. Overall, our work offers a significant contribution to the field of image encryption by providing a more efficient and secure approach that can be applied to a variety of image datasets.

The rest of the paper is structured as follows. Section 2 discusses the background work of different image encryption techniques. Section 3 discusses the proposed methodology. In Section 4, we present and discuss the metrics used for performance evaluation as well as the obtained numerical findings. Additionally, we do a comparative analysis with related image encryption techniques from the literature. The conclusion and future work of this paper is presented in Section 5.

2 Background work

There are many image encryption techniques have been proposed for exchanging the multimedia data as image

securely over a public channel with less computation cost. one of the early research publications (Matthews, 1989) put out the idea of creating a sequence that is both chaotic and unpredictable and used it in encryption techniques. To create a key stream that is also used in encryption, the author used a logistic chaotic map. A modified image encryption technique that is resistant to both plain text and selected plain text attack was proposed by Patidar et al. (2010). The encryption method is a chaotic substitution-diffusion that uses a chaotic logistic map and a chaotic standard. Because of this, it is robust and possesses the traits of both confusion and dispersion. Spatio-temporal dynamics, which combines linear and nonlinear coupled map systems, is implemented in Zhang and Wang (2014). This increases the coupling structure as a result. The parameters of the map and its beginning conditions use a key that is longer than 400 bits. The authors used bit-level pixel permutation. This makes it possible to manually permute the pixel's lower and higher bit planes. In order to encrypt greyscale images, Zhang (2018) used a cubic S-box and a piece-wise linear chaotic map to construct the key stream. The encryption and decryption procedures used in the implemented method are identical. An image algorithm based on DNA sequence operations was put into practice by Wu et al. (2018). The system parameters for the NCA map-based CML chaotic map, which generates the key, are updated using the SHA-256 hash function. After that, the RGB cover image's channels are split. Utilising DNA level merge-shuffle on each channel with a row-by-row permutation, the diffusion process is carried out. To increase the algorithm's robustness, a confusion process is then run using the key created. The encryption algorithm is resistant to common attacks. Hasanzadeh and Yaghoobi (2020) used fractal images, S-box, and hyper chaotic dynamic to construct an image encryption system. where four fractal images are initially created using the Julia fractal set keys. The values of the S-box created from the Hilbert fractal are used to replace the pixels of the cover image. Using the logistic map, the positions of the pixels are also moved around. The Chen hyper chaotic system is then used to choose 3 of the 4 fractal images before the shuffled image is split into its channels and XORed with them. A two-stage image encryption approach is suggested by Alexan et al. (2021). Rule 30 cellular automata are used in the first stage, and the Lorenz system of differential equations is solved in the second stage, which deals with a chaotic system. A PRNG key is applied and the plain image data is XORed at each stage. For greater bit-permutation, several cyclical shifts are applied in conjunction with the XORing process. Their system exhibits robustness against statistical attacks and a short encryption time, according to performance study. The confusion-diffusion picture encryption method was proposed by Elkandoz and Alexan (2022). Before being diffused by XORing its pixels with a secret key, the image's pixels are first shuffled after which they are first disarranged. A combination of various chaotic maps is used to produce this key.

The goal of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Typically only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that needs to somehow be explored. There are a number of algorithms (Su, 2014) present today that can be used for object detection.

The first successful algorithm was discriminatingly trained part-based models (DPM) (Felzenszwalb et al., 2010), which describes an object detection system based on mixtures of deformable part models. The problematic part for an object detector is the presence of lots of variations. These can arise from different viewpoints, non-rigid deformation and intra-class variability. The DPM tries to capture those variations. The Dalal and Triggs detector acts as a root filter and linear SVM is used for training in DPM. It assumes that an object is constructed from its various parts. The positions of the parts are not fixed and it penalises the parts that are far away from where they are supposed to be. Then, it uses a coarse root filter (standard HOG) and some higher resolution filter for the parts. So, based on a given location of the root it tries to find where the parts are. Then it sums the scores (based on the deformation) of all the parts to finally say that whether the thing is an object or not.

Integral channel features (ICF) (Dollar et al., 2009) is a method for object detection used in the field of computer vision. It uses integral images to extract features such as local sums, histograms and Haar-like features from multiple registered image channels. Once channels are obtained from an input image, various features can be extracted from these channels. These features are of two types, first-order channel features and higher-order channel features. The ChnFtrs method allows one to pool features that capture the richness from diverse channels. The authors used the first order features for their experimental results since the second order features were not adding much value. Training features by AdaBoost classifier, their method was fast and efficient on pedestration detection. They achieved an accuracy of 79% on INRIA dataset using PASCAL criteria. Convolution neural networks (CNN) (Krizhevsky et al., 2012) in terms of machine learning are feed forwarding artificial neural networks that are used to analyse visual imagery. CNNs can be used for object detection. Region-based convolutional neural networks (R-CNN) (Girshick et al., 2014) is an object detection system that is used for object detection in an image. It combines region proposals with rich features which are computed by the convolutional neural networks.

Duan et al. (2022) proposed the scheme that combines multi-parameter fractional discrete Tchebyshev moments (MPFrDTMs) with a nonlinear fractal permutation method. The MPFrDTMs are used to create a larger key space and withstand brute-force attacks, while the nonlinear fractal permutation method combines a fractal Sierpinski triangle model with rotation operations to resist known-plaintext and chosen-plaintext attacks. This scheme is shown to be feasible and highly secure based on simulation results and performance analyses.

Lai et al. (2023) proposed a novel chaos-based encryption scheme using a two-dimensional Salomon map to protect images from attacks. The scheme selectively exchanges high and low bits of the image and spreads altered pixels to random positions, resulting in a cipher image that only a unique key can recover. Comprehensive tests validate the scheme's cryptographic effect and security. The paper demonstrates the potential of using chaos-based systems for image encryption to protect valuable information.

The presented image encryption techniques have some limitations that need to be considered. One of the major limitations is that these techniques may not be suitable for encrypting specific objects, as they may not be able to handle complex shapes and textures. Another limitation is that statistical measures do not always produce better results, and the efficacy of these techniques is still in progress.

Furthermore, while these methods claim to be resistant to common attacks, their robustness and security against advanced attacks have not been thoroughly evaluated. It is essential to investigate the vulnerability of these techniques to advanced attacks to ensure their reliability in real-world scenarios.

Moreover, the computational cost and efficiency of these methods are not well-defined, and their applicability to large-scale image data is questionable. It is crucial to understand the computational requirements of these techniques and how they scale with large datasets. This information is necessary for deciding whether these methods are feasible and practical for real-world applications.

3 Proposed approach

Currently, there are no schemes in place that encrypt the main object before encrypting the entire image, and most available schemes deal with image encryption. However, as we all know, the most crucial element to hide in an image is the object. Therefore, we propose an approach in which we first hide the main object before encrypting the entire image with a key that depends on the image details.

The proposed encryption technique involves two phases to enhance the security of the object in the image. In the first phase, we locate the object in the image using object detection techniques and encrypt it using a key generated by an edge detection algorithm. In the second phase, we divide the image into clusters and encrypt each cluster separately using its cluster keys (index and c) to further increase its security. This approach ensures that even if the entire image is decrypted, the object will still be hidden, and only a legitimate user will be able to decrypt it. The following sections provide a detailed description of each phase.

3.1 Phase 1: detecting and encrypting the object

This phase involves object detection and encryption. First, we locate the object in the image using an object detection algorithm. We choose a window in the middle of the image, as most objects are present in this region. We use the Canny edge detector (Canny, 1987) to detect edges, but any other edge detector can be used (Davis, 1975). We set the intensity range to $[0.0001, 0.0005]$ with a standard deviation of 7.

Once the object is located, we encrypt it using a multiplicative cipher. The key for the cipher is generated using an edge detection algorithm that evaluates the key using the mean of all the edge pixel values in the image. This technique provides an additional layer of security and makes it difficult for an attacker to decipher the encrypted object. As a result, the object is successfully hidden from the rest of the image.

Algorithm 1 provides the pseudocode for generating keys required for encryption and decryption of the object. The input is the original image I of height h and width w, and the output is the keys s and s^{-1}.

Algorithm 1 Key generation algorithm

Input : Original image I
Output: Keys s and s^{-1}
Find edges in I using the Canny edge detector
Compute the sum S of intensities of all the edge pixels
$s \leftarrow (S \bmod 256)$
if $s < 120$ **then**
    Select n such that $s \leq n \leq 256$ and n has an inverse in $\mathbb{Z}_{256}$
    $s^{-1} \leftarrow (n^{-1} \bmod 256)$
else
    Select n such that $1 \leq n \leq s$ and n has an inverse in $\mathbb{Z}_{256}$
    $s^{-1} \leftarrow (n^{-1} \bmod 256)$
return s, s^{-1}

The algorithm begins by detecting edges in the original image I using the Canny edge detector. It then computes the sum S of intensities of all the edge pixels. The key s is obtained by taking the modulus of S with 256.

The next step is to compute the inverse key s^{-1}, which will be used for decrypting the object. If s is less than 120, then n is selected such that $s \leq n \leq 256$ and n has an inverse in $\mathbb{Z}_{256}$. The inverse of n is then computed, and stored as s^{-1}. Otherwise, if s is greater than or equal to 120, n is selected such that $1 \leq n \leq s$ and n has an inverse in $\mathbb{Z}_{256}$. The inverse of n is then computed, and stored as s^{-1}.

Finally, the algorithm returns both keys s and s^{-1} for use in the encryption and decryption of the object.

Algorithm 2 provides the pseudocode for object encryption using a multiplicative cipher with key s, where T is the object-encrypted image.

Here, p_i and p_j are the indices of the pixel in the random permutation, and s is the key for encryption. The output T is the object-encrypted image.

In this algorithm, the object-encrypted image T is first input and the output includes the keys s_1 and ${s_1}^{-1}$, as well as the noisy image N. The Canny edge detector is used to find the edges in T, and the sum S of all the intensities of

the edges is computed. The key s_1 for encryption is then generated as $s_1 \leftarrow (S \bmod 256)$. The inverse key ${s_1}^{-1}$ is generated by selecting a number n such that $s_1 \leq n \leq 256$ and n has an inverse in $\mathbb{Z}_{256}$ if $s_1 < 120$, and $1 \leq n \leq s_1$ and n has an inverse in $\mathbb{Z}_{256}$ otherwise. Finally, the whole image is encrypted by multiplying each pixel value in T by s_1 and taking the result modulo 256 to get the corresponding pixel value in N.

Algorithm 2 Algorithm for object encryption

Input : Original image to encrypt: I, Keys for encryption: s and s^{-1}
Output: Object-encrypted image: T
Create a random permutation of pixels; **for** $i \leftarrow 1$ *to* w **do**
  for $j \leftarrow 1$ *to* h **do**
    $T(p_i, p_j) \leftarrow (I(p_i, p_j) \times s) \mod 256$;

Algorithm 3 Algorithm for generating keys s_1 and ${s_1}^{-1}$, and encrypting the whole image

Input : Object-encrypted image T
Output: Keys s_1 and ${s_1}^{-1}$, and noisy image N
Find edges using the Canny edge detector on T
Compute the sum S of all the intensities of the edges
$s_1 \leftarrow (S \bmod 256)$
if $s_1 < 120$ **then**
  Select n such that $s_1 \leq n \leq 256$ and n has an inverse in $\mathbb{Z}_{256}$
  $s1^{-1} \leftarrow (n^{-1} \bmod 256)$
else
  Select n such that $1 \leq n \leq s_1$ and n has an inverse in $\mathbb{Z}_{256}$
  ${s_1}^{-1} \leftarrow (n^{-1} \bmod 256)$
for $i \leftarrow 1$ *to* h **do**
  for $j \leftarrow 1$ *to* w **do**
    $N_{i,j} \leftarrow (T_{i,j} \times s_1) \mod 256$
return s_1, ${s_1}^{-1}$, N

3.2 Phase 2: encryption using K-means clustering

This phase involves dividing the output image of phase 1 into a number of clusters and encrypting them individually. In the proposed algorithm, we use K-means clustering for this purpose. The input to K-means clustering is a column matrix of image pixels. After applying K-means clustering, we get two matrices as output: one is cluster indices (idx) and the other is cluster centroids (c). Cluster indices provide information about which pixel belongs to which cluster. The cluster centroids provide information about each cluster centre and will be used for generating a cluster key and its inverse key for each particular cluster.

The cluster key and its inverse key are generated using the mean of that cluster, using Algorithm 4. k_i is the key that is used to encrypt cluster i, and k_i^{-1} is the inverse of the key that is used to decrypt the encrypted cluster. (idx) is the cluster indices vector.

Algorithm 4 Algorithm for generating keys k_i and k_i^{-1} for each cluster

Input : Phase 1 encrypted image N, Cluster indices vector (idx), Number of clusters k
Output: Vector k and k^{-1} for each cluster
Let c be the matrix of cluster centroids; Let x be the vector of mean values of the clusters; **for** $i \leftarrow 1$ *to* k **do**
  if $x_i < 120$ **then**
    Select n from Z_{256} such that $x_i \leq n \leq 256$ and its inverse exists in Z_{256}; $k_i^{-1} \leftarrow (n^{-1} \bmod 256)$;
  else
    Select n from Z_{256} such that $1 \leq n \leq x_i$ and its inverse exists in Z_{256}; $k_i^{-1} \leftarrow (n^{-1} \bmod 256)$;
  $k_i \leftarrow c_i$;

3.3 Phase 2: encryption using K-means clustering

This phase involves dividing the output image of phase 1 into a number of clusters and encrypting them individually. In the proposed algorithm, we use K-means clustering for this purpose. The input to K-means clustering is a column matrix of image pixels. After applying K-means clustering, we get two matrices as output: one is cluster indices (idx) and the other is cluster centroid (c). Cluster indices provide information about which pixel belongs to which cluster. The cluster centroid provides information about each cluster centre and will be used for generating a cluster key and its inverse key for every particular cluster.

Algorithm 5 Algorithm to encrypt using K-means clustering

Input : phase 1 encrypted image N, cluster indices: idx, cluster centroids: c, cluster keys: k_i
Output: An encrypted image E
reshape the input image into a column matrix; **for** $i \leftarrow 1$ *to* $h \times w$ **do**
  $j \leftarrow idx(i)$; `// get the cluster index for the current pixel`
  $E(i) \leftarrow (N(i) \times k_j) \mod 256$

The cluster key and its inverse key are generated using the mean of that cluster, using Algorithm 4. k_i is the key that is used to encrypt cluster i, and k_i^{-1} is the inverse of the key that is used to decrypt the encrypted cluster. (idx) is the cluster indices vector.

Note: the algorithm in phase 2 assumes that the number of clusters k is already known. In practice, the number of clusters can be determined using techniques such as the elbow method or the silhouette method.

3.4 Encrypting the keys generated in phases 1 and 2

After the above phases are done, we need to encrypt the keys that were generated during phases 1 and 2. In phase 1, we used 2 keys to encrypt the objects, as shown in Algorithm 2.

For the phase 1 keys, we can share a key using a station-to-station protocol (Desmedt, 2011) and then encrypt the keys using a quadratic function taking the shared key as its value and the two keys as its roots.

Algorithm 6 Algorithm for encrypting phase 1 keys

Input : Phase 1 inverse keys s^{-1} and s_1^{-1}
Output: Y and $(-K_1 \times K_2)$
$K1 \leftarrow s^{-1}$
$K2 \leftarrow s_1^{-1}$
Make one of the keys negative.
$Y \leftarrow X^2 - (-K_1 + K_2) \times X + (-K_1 \times K_2)$
return Y and $(-K_1 - K_2)$

Here, both Y and $(-K_1 \times K_2)$ will be sent to the receiver to obtain the original keys.

For phase 2, we need to encrypt the cluster indices and inverse keys that are used to encrypt the image. Even if an attacker obtains the cluster indices, they cannot determine which key is used for each cluster. We can add the value of the cluster indices to their corresponding encrypted image pixels, encrypt the result using a key, and then add it to the encrypted image.

Algorithm 7 Algorithm for decryption

Input : Encrypted image E and inverse keys k^{-1}
Output: Decrypted image: D
Reshape E into a column matrix of height $h \times w$ Let J be a column matrix of zeros with the same size as E **for** $i \leftarrow 1$ *to size of* k^{-1} *vector* **do**
  $J(idx == i) \leftarrow (E(idx == i)) \times k^{-1}(i) \mod 256$
Reshape J to an image of height h and width w Let X be a variable image $X \leftarrow (J \times s_1^{-1}) \mod 256$ **for** $i \leftarrow 1$ *to* h **do**
  for $j \leftarrow 1$ *to* w **do**
    $D(i,j) \leftarrow (X(i,j) \times s^{-1}) \mod 256$

3.5 Decryption

Encrypted image will be decrypted by using the inverse keys of the different phases but in reverse steps. For decryption, first we have to apply inverse keys, i.e., decrypting the cluster indices and then using them to decrypt the clusters with their respective inverse keys. After phase 2 decryption, phase 1 decryption will be done using the inverse keys of phase 1. The same procedure will be followed for decryption as in encryption but with inverse keys. Algorithm 7 defines the procedure for decryption.

Here, $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$ denote the coordinates of the object window in the image.

4 Experimental results and analysis

The experimental tests were run on Intel Core(TM) i7-7700 HQ CPU @ 2.8 GHz and 8 GB of DDR 4 @ 2,400 MHz RAM. The time consumed with 11 clusters took about 2.3 seconds. The results of our proposed algorithm is calculated by taking a number of images as an input. We have experimented on different greyscale images with different dimensionality. In this paper, we have shown the results of four different greyscale images. The experiment was conducted in the ECCSD Dataset (2023) which contains 1,000 images, for illustration we have used four sample images.

Figure 1 Experimental result of proposed scheme, (a–d) original images (e–h) object encrypted images (i–l) encrypted images (m–p) decrypted image (see online version for colours)

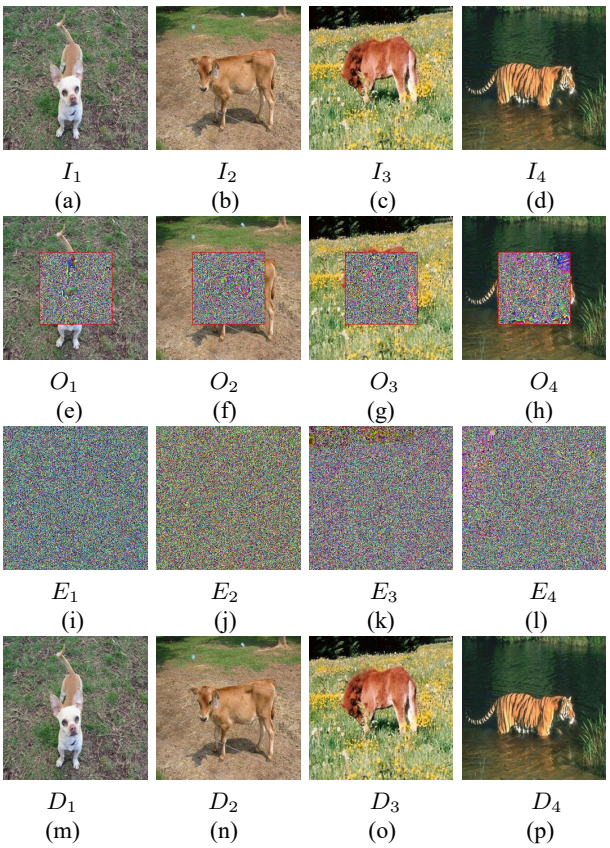
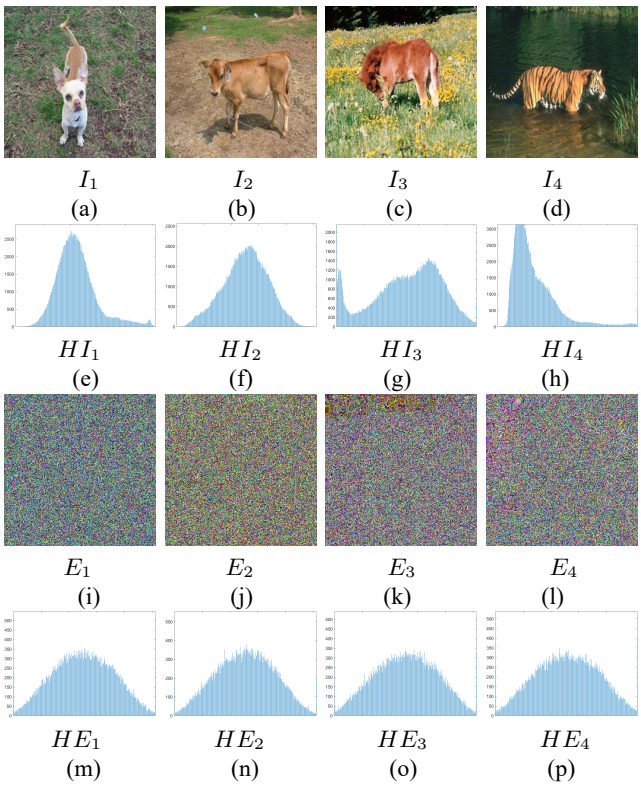


Figure 1 shows the experiment result of proposed image and object encryption scheme. The experimental results of proposed scheme is shown in Figure 1. The original images I_1, I_2, I_3 and I_4 are depicted in Figures 1(a)–1(d), the object encrypted images O_1, O_2, O_3 and O_4 are depicted in Figures 1(e)–1(h), the encrypted images E_1, E_2, E_3 and E_4 are depicted in Figures 1(i)–1(l), the decrypted images D_1, D_2, D_3 and D_4 are depicted in Figures 1(m)–1(p). The outcome demonstrates that the encrypted images are completely randomised and do not reveal original image information as well as decrypted image is lossless in nature, i.e., it is same as original image.

Figure 2 Experimental result of proposed scheme, (a–d) original images (e–h) histogram of original images (i–l) encrypted images (m–p) histogram of encrypted image (see online version for colours)



4.1 Histogram analysis

Histograms are frequently used to show how the pixel values in an image are distributed overall. A natural meaningful image's histogram typically has clear statistical properties. If the encryption procedure is effective, the encrypted image's histogram should have a uniform distribution. It is evident from Figure 2 that the histogram of the encrypted image has a uniform distribution. In other words, even if an attacker gets a hold of a plain image, they are unable to decipher the encrypted image. The first column shows four different original images, second column shows the histogram of respective original images, third column shows encrypted images and fourth column shows histogram of respective encrypted images. The original images I_1, I_2, I_3 and I_4 are depicted in Figures 2(a)–2(d), the original images' histograms HI_1, HI_2, HI_3 and HI_4 are depicted in Figures 2(e)–2(h), the Encrypted images E_1, E_2, E_3 and E_4 are depicted in Figures 2 (i)–2(l), the Encrypted images' histograms HE_1, HE_2, HE_3 and HE_4 are depicted in Figures 2(m)–2(p). If an attack occurs, it will be challenging to decipher the encrypted image because of the histogram's nearly uniform distribution.

4.2 Statistical analysis

Statistical analysis helps to analyse the security and efficiency of image encryption algorithm. The significant statistical measures are used to evaluate the statistical similarity between the images, such as peak signal to noise ratio (PSNR), correlation, root mean square error (RMSE). The number of pixel change rate (NPCR), unified averaged changed intensity (UPCI), and information entropy, are used to check the strength of an algorithm and robustness of an algorithm.

1 *Correlation:* The correlation coefficient is used to identify the similarity between two images. The correlation coefficient value ranges from -1 to 1. A correlation coefficient value close to 1 represents positive correlation, -1 close to negative correlation and close to 0 no correlation between the two images. The correlation coefficient between two images s, t is evaluated using equation (2).

$$P_{s,t} = \frac{\frac{1}{M}\sum_{i=1}^{M}(a_i - \bar{s})(t_i - \bar{t})}{\sqrt{\left(\frac{1}{M}\sum_{i=1}^{M}(s_i - \bar{s})^2\right)\left(\frac{1}{M}\sum_{i=1}^{M}(t_i - \bar{t})^2\right)}} \tag{2}$$

where $\bar{s}$, $\bar{t}$ represent the mean values of s and t respectively.

2 *RMSE:* It is used to determine the similarity between two images A, B. The similarity between the two images is inversely proportional to the RMSE value. RMSE between two images is evaluated using equation (3).

$$RMSE = \sqrt{\frac{1}{(s \times t)}\sum_{i=1}^{s}\sum_{j=1}^{t}(A(i,j) - B(i,j))^2} \tag{3}$$

where s and t are the dimensions of an image.

3 *PSNR:* It is used to measure the quality of image. PSNR value is evaluated using equation (4).

$$PSNR = 10\log_{10}\left(\frac{255 \times 255}{(MSE)}\right) \tag{4}$$

where MSE is the mean square error, and 255 is the highest intensity in the image. In general, if the PSNR value of the share and cover image is greater than 30 dB, it is difficult for the human eye to distinguish between the two images.

4 *NPCR:* The average number of intensity changes and pixel changes are tested by the NPCR. Using equation (5), the NPCR value is assessed.

$$NPCR(I,E) = \frac{\sum_{i,j} F(i,j)}{(x \times y)} \times 100\% \quad (5)$$

where I is the original image and E is the encrypted image, and x, y are the dimension of the image. The $F(i,j)$ value is evaluated using equation (6).

$$F(i,j) = \begin{cases} 1, & \text{if } I(i,j) \neq E(i,j). \\ 0, & \text{if } I(i,j) = E(i,j). \end{cases} \quad (6)$$

5 *UACI:* The UACI is used to assess the algorithm's resistance to differential attack. Using equation (7), the UACI value is assessed.

$$UACI = \left[\frac{1}{(x \times y)} \sum_{i,j} \frac{|I(i,j) - E(i,j)|}{255} \right] \quad (7)$$

where (x, v) represents the dimension of the image, 255 is highest intensity value in the image.

6 *Information entropy:* The information entropy is used to quantify the randomness of encrypted images. The highly secure technique that hides the information is chosen based on the encrypted image value being close to the ideal value of 8. Equation (8) is used to visualise the information entropy.

$$IE = -\sum_{i=0}^{2^M - 1} P(E_i) \log_2(P(E_i)) \quad (8)$$

where M is number of bits to represent the pixel, E_i represents the encrypted image, $P(E_i)$ represents the probability of the pixel.

Table 1 shows the correlation between adjacent pixels in the different horizontal, vertical, and diagonal orientations for various original and corresponding encrypted images. The correlation coefficient of the original image is close to 1, while that of the encrypted image is not that much close to 0, indicating that the pixels are correlated with one another, rendering the encrypted image easy to attack, However, because to the two-level encryption, the attacker will not be able to obtain the original object encrypted inside the image.

Table 2 shows statistical comparison based on RMSE, and PSNR of four different images with respective encrypted images. A PSNR value of less than 27 dB indicates that the original and encrypted images have dissimilarities in terms of visual quality and the higher RMSE value indicates the dissimilarity between the original and encrypted images. The results show that the proposed scheme gives almost the same result as the existing schemes.

The statistical comparisons between the original and encrypted images are shown in Table 3. The encrypted images are very random and secure against differential attacks, as indicated by the NPCR value being higher than 99. The resulting encrypted images are randomised and secure against differential attacks, according to the UACI value higher than 33. The results show that the proposed scheme gives almost the same result as the existing schemes.

Table 1 Correlation coefficients of original and encrypted images

	Original image			*Encrypted image*		
	Horizontal	*Diagonal*	*Vertical*	*Horizontal*	*Diagonal*	*Vertical*
I_1	0.7708	0.6875	0.7969	0.0350	0.0162	0.0557
I_2	0.8902	0.8142	0.8679	0.0288	0.0185	0.0495
I_3	0.9024	0.8551	0.9294	0.0152	0.0260	0.0669
I_4	0.8942	0.8714	0.9601	0.1260	0.0565	0.0928

Table 2 Statistical comparison based on RMSE, and PSNR of four different images with respective encrypted images

	RMSE		
	Alexan et al. (2021)	*Elkandoz and Alexan (2022)*	*Proposed*
I_1, E_1	72.56	72.56	74.20
I_2, E_2	70.31	70.31	66.84
I_3, E_3	73.39	73.39	78.97
I_4, E_4	80.36	80.36	86.38
	PSNR (dB)		
	Alexan et al. (2021)	*Elkandoz and Alexan (2022)*	*Proposed*
I_1, E_1	10.91	10.91	10.72
I_2, E_2	11.19	11.19	11.62
I_3, E_3	10.81	10.81	10.18
I_4, E_4	10.03	10.03	9.37

The statistical measurements for a pair of original and decrypted images are shown in Table 4. The original and decrypted images are identical, as indicated by the correlation value of 1. The values of RMSE, UACI, and NPCR are 0, indicating that there is no error and that the encrypted images have been perfectly recovered. The PSNR score of ∞ indicates that the visual quality of the original and encrypted images is identical.

Table 5 represents the encrypted images E_1 to E_4 with information entropy values that are close to 8. This indicates that the encrypted images produced by the method would hardly expose the information. The strategy is therefore quite secure. The results show that the existing scheme (Elkandoz and Alexan, 2022) is almost equivalent or has better security than the proposed scheme, while Duan et al. (2022) have almost equivalent to proposed scheme.

Table 3 NPCR and UACI of four different images with respective decrypted images

	NPCR					*UACI*				
	Ben Slimane et al. (2018)	*Elkandoz and Alexan (2022)*	*Duan et al. (2022)*	*Lai et al. (2023)*	*Proposed*	*Ben Slimane et al. (2018)*	*Elkandoz and Alexan (2022)*	*Duan et al. (2022)*	*Lai et al. (2023)*	*Proposed*
I_1, E_1	99.62	99.60	99.62	99.60	99.61	33.45	32.22	33.54	33.47	33.49
I_2, E_2	99.62	99.62	99.58	99.58	99.59	33.47	30.98	33.56	33.45	33.47
I_3, E_3	99.60	99.60	99.61	99.59	99.67	33.48	30.53	33.46	33.47	33.53
I_4, E_4	99.62	99.61	99.62	99.61	99.62	33.47	29.60	33.43	33.45	33.51

Table 4 Quantitative analysis of four different images with respective decrypted images

	RMSE	*PSNR*	*Correlation*	*NPCR*	*UACI*
I_1, D_1	0	∞	1.00	0	0
I_2, D_2	0	∞	1.00	0	0
I_3, D_3	0	∞	1.00	0	0
I_4, D_4	0	∞	1.00	0	0

Table 5 Information entropy measurements of various encrypted images of the proposed and existing models

Encrypted images	*Information entropy*		
	Elkandoz and Alexan (2022)	*Duan et al. (2022)*	*Proposed*
E_1	7.1321	5.49	7.7444
E_2	7.1142	7.10	7.7148
E_3	7.1957	7.33	7.7708
E_4	7.1126	7.27	7.7850

5 Conclusions and future work

The proposed image and object encryption approach using a multiplicative cipher and K-means clustering algorithm is an effective and secure method for securing sensitive data contained within images. While the algorithm does employ computationally intensive techniques such as object detection and K-means clustering, it is still relatively fast, with the time taken to encrypt the image increasing linearly with the size of the image. The technique is lossless and carefully selects encryption parameters and reverses the encryption steps during decryption, ensuring that the original data is not distorted or degraded. The proposed approach performs comparably to existing schemes, as shown by histogram analysis and different statistical measures such as RMSE, PSNR, Correlation, NPCR, UACI, and information entropy. The security provided by the different statistical measures, such as correlation, PSNR, and RMSE, demonstrates the difficulty of attack and the effectiveness of the proposed approach. The UACI and NPCR values suggest that the proposed approach is difficult to attack. The information entropy value close to 8 signifies that the strength of the algorithm is better against attack. The histogram's almost uniformly distributed data suggests that encrypted images are completely randomised. Additionally, the proposed approach provides double security, making it more secure than existing schemes, as demonstrated through information entropy and different statistical measures. The approach is resistant to both statistical and differential attacks, making it suitable for a wide range of industrial applications. While the proposed technique uses a fixed-sized window at the centre of the image for object detection, it can be easily extended to encrypt salient objects located anywhere in the image. Overall, the experimental results demonstrate the efficacy of the proposed approach, and it has the potential to be a valuable tool in the field of image and data security.

References

Alexan, W., ElBeltagy, M. and Aboshousha, A. (2021) 'Lightweight image encryption: cellular automata and the Lorenz system', *2021 International Conference on Microelectronics (ICM)*, IEEE, pp.34–39.

Arab, A., Rostami, M.J. and Ghavami, B. (2019) 'An image encryption method based on chaos system and AES algorithm', *The Journal of Supercomputing*, Vol. 75, No. 10, pp.6663–6682.

Ben Slimane, N. et al. (2018) 'A novel chaotic image cryptosystem based on DNA sequence operations and single neuron model', *Multimedia Tools and Applications*, Vol. 77, No. 23, pp.30993–31019.

Borji, A. et al. (2014) *Salient Object Detection: A Survey*, arXiv preprint, arXiv:1411.5878.

Burkardt, J. (2009) *K-Means Clustering*, Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics.

Canny, J. (1987) 'A computational approach to edge detection', *Readings in Computer Vision*, Vol. 8, No. 6, pp.184–203.

Davis, L.S. (1975) 'A survey of edge detection techniques', *Computer Graphics and Image Processing*, Vol. 4, No. 3, pp.248–270.

Desmedt, Y. (2011) 'Station-to-station protocol', *Encyclopedia of Cryptography and Security*, pp.1256–1256, Springer, USA.

Dollar, P. et al. (2009) 'Integral channel features', *British Machine Vision Conference*, p.91-1 https://api.semanticscholar.org/CorpusID:14924524.

Duan, C-F. et al. (2022) 'New color image encryption scheme based on multi-parameter fractional discrete Tchebyshev moments and nonlinear fractal permutation method', *Optics and Lasers in Engineering*, Vol. 150, No. 7, p.106881.

ECCSD Dataset [online] https://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html (accessed 21 February 2023).

Elkandoz, M.T. and Alexan, W. (2022) 'Image encryption based on a combination of multiple chaotic maps', *Multimedia Tools and Applications*, Vol. 81, No. 18, pp.1–22.

Felzenszwalb, P.F. et al. (2010) 'Object detection with discriminatively trained part-based models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp.1627–1645.

Forouzan, B.A. and Mukhopadhyay, D. (2011) *Cryptography and Network Security (Sie)*, McGraw-Hill Education, West Patel Nagar, New Delhi.

Girshick, R. et al. (2014) 'Rich feature hierarchies for accurate object detection and semantic segmentation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Gupta, M. et al. (2022) 'An efficient image encryption technique based on two-level security for internet of things', *Multimedia Tools and Applications*, Vol. 82, No. 3, pp.1–21.

Han, J., Pei, J. and Kamber, M. (2011) *Data Mining: Concepts and Techniques*, Elsevier, Waltham, Massachusetts, USA.

Hasanzadeh, E. and Yaghoobi, M. (2020) 'A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys', *Multimedia Tools and Applications*, Vol. 79, No. 11, pp.7279–7297.

Hore, A. and Ziou, D. (2010) 'Image quality metrics: PSNR vs. SSIM', *2010 20th International Conference on Pattern recognition (ICPR)*, IEEE.

Jain, A.K. (1989) *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., USA.

Jähne, B. (2012) *Digital Image Processing*, Springer Publishing Company, Incorporated, Heidelberg, Germany.

Kahate, A. (2013) *Cryptography and Network Security*, Tata McGraw-Hill Education, 7 West Patel Nagar, New Delhi.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'ImageNet classification with deep Convolutional neural networks', *Advances in Neural Information Processing Systems*, Vol. 25, pp.1–9.

Lai, Q. et al. (2023) 'A novel pixel-split image encryption scheme based on 2D Salomon map', *Expert Systems with Applications*, Vol. 213, No. Part A, p.118845.

Lehmann, E.L. and Casella, G. (2006) *Theory of Point Estimation*, Springer Science & Business Media, New York, USA.

Lu, Q., Zhu, C. and Deng, X. (2020) 'An efficient image encryption scheme based on the LSS chaotic map and single S-box', *IEEE Access*, Vol. 8, pp.25664–25678.

Luo, Y. et al. (2019) 'An image encryption method based on elliptic curve elgamal encryption and chaotic systems', *IEEE Access*, Vol. 7, pp.38507–38522.

Man, Z. et al. (2021) 'Double image encryption algorithm based on neural network and chaos', *Chaos, Solitons & Fractals*, Vol. 152, p.111318.

Matthews, R. (1989) 'On the derivation of a chaotic encryption algorithm', *Cryptologia*, Vol. 13, No. 1, pp.29–42.

Patidar, V. et al. (2010) 'Modified substitution–diffusion image cipher using chaotic standard and logistic maps', *Communications in Nonlinear Science and Numerical Simulation*, Vol. 15, No. 10, pp.2755–2765.

Rabbani, M. (2002) 'JPEG2000: image compression fundamentals, standards and practice', *Journal of Electronic Imaging*, Vol. 11, No. 2, p.286.

Rawat, A. and Deshmukh, M. (2020a) 'Communication efficient Merkle-tree based authentication scheme for smart grid', *IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pp.693–698.

Rawat, A. and Deshmukh, M. (2020b) 'Tree and elliptic curve based efficient and secure group key agreement protocol', *Journal of Information Security and Applications*, Vol. 55, p.102599.

Rawat, A.S. and Deshmukh, M. (2019) 'Efficient extended diffie-hellman key exchange protocol', *International Conference on Computing, Power and Communication Technologies (GUCON)*, IEEE, pp.447–451.

Rawat, A.S. and Deshmukh, M. (2021a) 'Computation and communication efficient Chinese remainder theorem based multi-party key generation using modified RSA', *Security and Privacy: Select Proceedings of ICSP 2020*, pp.25–32.

Rawat, A.S. and Deshmukh, M. (2021b) 'Computation and communication efficient secure group key exchange protocol for low configuration system', *International Journal of Information Technology*, Vol. 13, No. 3, pp.839–843.

Stallings, W. (2006) *Cryptography and Network Security: Principles and Practices*, Pearson Education, India.

Su, J-C. (2014) *State-of-the-Art Object Detection Algorithms*, University of California, San Diego, Vol. 9500 [online] https://api.semanticscholar.org/CorpusID:203600435.

Wang, X., Feng, L. and Zhao, H. (2019) 'Fast image encryption algorithm based on parallel computing system', *Information Sciences*, Vol. 486, No. C, pp.340–358.

Wu, X. et al. (2018) 'Color image DNA encryption using NCA map-based CML and one-time keys', *Signal Processing*, Vol. 148, pp.272–287

Yoneyama, S. and Murasawa, G. (2009) 'Digital image correlation', *Experimental Mechanics*, Vol. 207, pp.1–10.

Zhang, Y. (2018) 'The unified image encryption algorithm based on chaos and cubic S-box', *Information Sciences*, Vol. 450, No. C, pp.361–377.

Zhang, Y-Q. and Wang, X-Y. (2014) 'A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice', *Information Sciences*, Vol. 273, pp.329–351.