scientific reports

Check for updates

OPEN

Efficient three layer secured adaptive video steganography method using chaotic dynamic systems

D. Kumar[1], V. K. Sudha[2], N. Manikandan[3] & Krishnaraj Ramaswamy[4]✉

In light of the unprecedented growth in internet usage, safeguarding data from unauthorized access has emerged as a paramount concern. Cryptography and steganography stand as pivotal methods for ensuring data security during transmission. This study introduces an innovative adaptive video steganography approach featuring three tiers of security for extracting concealed information, thereby facilitating secure communication. The embedding process operates within the spatial domain of cover video frames, enabling a remarkable hiding ratio of up to 28.125% (equivalent to 2.25 bits per pixel in payload) without compromising the quality of video frames. Users are afforded the flexibility to select between partial or full embedding capacity of C_{VF} through the proposed adaptive control block (ACB). The chaotic key generator (CKG), which combines a logistic map and sine map, is employed to generate highly sensitive initial seeds for permutation order (PO), frame selection (FS), and random position for hiding (RPH), thereby ensuring three levels of security. Prior to transmission, both C_{VF} and hidden data (S_D) are encrypted using PO. Encrypted C_{VFs} are then randomly selected using FS for embedding, with RPH employed during the embedding process. Subsequently, for transmitting the stego-video frame, embedded C_{VFs} are decrypted using the same PO. Experimental results demonstrate the efficacy of the proposed approach, achieving an adaptive hiding ratio ranging from 7.1 to 28.125% (equivalent to 0.56 to 2.25 bits per pixel in payload) and maintaining a peak signal-to-noise ratio (PSNR) within the range of 50.25 to 62.05 dB.

Keywords Adaptive video steganography, Chaotic key generator, Hiding ratio, Payload, PSNR

The significant growth of the internet has enabled users to transmit massive amounts of personal data over the network, particularly in the form of video. As a result, protecting data that is shared over the internet is critical[1]. Consequently, cryptographic techniques can be used to transfer data in a secured manner. It is a technique for representing data in an irrelevant and difficult-to-understand format[1,2]. Despite the fact that many cryptographic methods for encrypting and decrypting data have been developed, these methods are inefficient due to the rapid growth of the internet[11,12]. The irrelevant form of cipher text in cryptography will pique the intruder's interest. This issue can be elucidated using a technique known as steganography. Steganography is the technique of concealing secret data in order to prevent the disclosure of a hidden message[13]. The cover, along with the secret data, is referred to as stego data. The cover and secret data can take many forms, including text, audio, video, and image.

Three key measures of a steganography system effectiveness are PSNR, Payload, and Hiding Ratio[14,15]. The PSNR is used to compare the similarity of the cover frame and the stego frame[16]. The payload[14] represents the number of bits allocated for embedding the secret data in the cover frame using the steganographic technique and the hiding ratio[16] evaluates the space allocated for the secret data in the cover frame in terms of percentage. The primary goal of steganography is to avoid suspicion in the transmission of a hidden message. If there is any suspicion, the steganography algorithm is useless. The majority of researchers are drawn to image steganography[12,13,15,17]. It employs an image as the cover and allots more space for embedding the secret data. Video steganography is a more advanced version of image steganography that can allocate even more space than image steganography. Because of this benefit, video steganography has grown in popularity[14,18].

[1]P.A. College of Engineering and Technology, Pollachi, India. [2]Dr. Mahalingam College of Engineering and Technology, Pollachi, India. [3]P.A. College of Engineering and Technology, Pollachi 642002, India. [4]Department of Mechanical Engineering, College of Engineering Science, Dambi Dollo University, Dembi Dolo, Ethiopia. ✉email: dr.krishnarajdirectorcei@dadu.edu.et

Related works

In the Kacar et al.[4] method, video frames are randomly selected using values from a 4D chaotic system to conceal secret data. This approach achieves a PSNR of 55 dB with a payload of 0.64 bpp, resulting in a hiding ratio of 8.08%. However, this method only offers single-layer security. Darani et al.[5] developed an image steganography algorithm for embedding gray scale secret image into RGB color image, by employing single layer security through chaotic maps and genetic algorithm. The method achieves PSNR of 47.88Db for a payload of 0.2222bpp, corresponding to hiding ratio of 2.77%. Aparna and Madhumitha[6] proposed combined image encryption and steganography algorithm based on least significant substitution (LSB) method to embed the secret data into the cover using multiple chaotic maps. The method achieves PSNR of 56.10 dB. Zakaria et al.[7] proposed data hiding approach based on LSB substitution using a mapping bits' strategy. The method achieves a PSNR of 42.15 dB for a payload of 3.24bpp, corresponding to hiding ratio of 40.5%.

Zhang and Chen[18], proposed an H.264/AVC intra prediction Mode (IPM) video steganography algorithm based on (n, k) linear block code. The algorithm uses I4 video frame blocks to embed secret messages with a payload of 1.33bpp or a hiding ratio of 16.625%. Younus and Hussain[13], proposed an image steganography algorithm that combines cryptography and steganography techniques. To increase the security and payload, the secret image is encrypted and compressed using the Vigenere cipher and Huffman coding. The method achieves a PSNR of 55.71 dB for a payload of 1.59bpp, corresponding to a hiding ratio of 19.875%. Narayanan et al.[19], proposed a video steganography algorithm based on the least significant bit substitution (LSB) method, which uses a 3D chaotic map to select pixels in a video. To encode data to be transmitted into pixels, a 3–3–2 substitution method is used, which means that the LSB of all RGB color components is taken (3 bits of red, 3 bits of green and 2 bits of blue). Kar and Mandal[16], proposed a DNA-based video steganography algorithm that uses the least significant bit substitution method. The method achieves PSNRs ranging from 46.21 to 52.24 dB for payloads ranging from 1.23 to 1.78 bpp or hiding ratios ranging from 15.375 to 22.25%. Based on a human vision region of interest and a face detection algorithm Balu et al.[20], proposed a video steganography algorithm in medical imaging system. To increase security, the method embeds secret information in different levels based on human visual region of interest. It achieves a PSNR of 67.17 dB with a payload of 0.1 bpp for hiding ratio of 1.25%. Abed et al.[21], proposed a two-level security-based video steganography method based on the LSB. A 1–1–0 LSB technique is used to hide secret data in video files, which means that they take the LSB of RG components (1bit of red and 1 bit of green). The method achieves a PSNR of 57.1 dB. Mstafa et al.[22], proposed a video steganography algorithm based on multiple object tracking (MOT) and error correcting codes (ECC) in the transform domain, like DWT and DCT. The method achieves a PSNR of 49.01 dB for DWT and 48.67 dB for DCT with payloads of 0.27bpp and 0.28bpp, respectively and a hiding ratio of 3.4% for DWT and 3.46% for DCT. Mstafa and Elleithy[23], proposed a wavelet-based video steganography algorithm based on the KLT tracking algorithm and BCH codes. The secret image is embedded in the LH, HL, and HH coefficients of all facial regions of video frames by this algorithm. The method achieves a PSNR of 41–50 dB for a payload of 0.35bpp or a hiding ratio of 4.4%. Kelash et al.[24], used color histograms to directly embed data into video frames, where each pixel in each video frame is divided into two parts, and the number of bits that will be embedded in the right side part is counted in the left side part. The method achieves a PSNR of 48 dB for a payload of 0.09bpp or a hiding ratio of 0.6%. Alavianmehr et al.[25], proposed a robust lossless video steganographic method based on histogram distribution constraints (HDC). The method embeds the secret data in the video frame's luminance (Y) component. It achieves a PSNR of 36.64 to 36.97 dB for a payload of 1 or a hiding ratio of 12.5%. Hu and Tak[26], proposed a method for video steganography based on non-uniform rectangular partitioning. In this secret video stream, at least four significant bits of each frame of the cover video with nearly the same size are hidden. PSNR is in the 28.19—29.75 dB range. Ranjithkumar et al.[14], proposed video steganography method based chaos. The method achieves PSNR of 49 dB for a payload of 2 or hiding ratio of 25%.

All of the similar methods discussed above use video as a cover. There is always tradeoff between PSNR and payload. Younus and Hussain[13], achieved a PSNR of 55.71 dB for hiding ratio of 19.88%, which is less than Ranjithkumar et al.[14], Kar et al.[16], Hu and Tak[26], Kacar et al.[4], Darani et al.[5], Zakaria et al.[7].

Balu et al.[20], achieved a PSNR of 67.12 dB for hiding ratio 1.25%, which is very less than Younus et al.[13], Ranjithkumar et al.[14], Kar et al.[16], Mstafa et al.[22], Alavianmehr et al.[25], Hu et al.[26] With hiding ratio of 22.25% Kar et al.[16], achieved PSNR of 52.24 dB which is less than[13,20]. For payload of 12.5% Alavianmehr et al.[25] achieves PSNR of 36.97 dB which is very less than Younus et al.[13], Ranjithkumar et al.[14], Kar et al.[16], Balu et al.[20] and Mstafa et al.[22].

Mstafa et al.[22], achieved PSNR of 49.01 dB for hiding ratio is 3.46%, which is less than Younus and Hussain[13], Ranjithkumar et al.[14], Kar et al.[16], Balu et al.[20], Alavianmehr et al.[25] and Hu and Tak[26] With hiding ratio of 50% Hu and Tak[26], achieve a PSNR of 29.75 dB which is significantly lower than Younus et al.[13], Ranjithkumar et al.[14], Kar et al.[16], Balu et al.[20], Mstafa et al.[22], and Alavianmehr et al.[25], and so the stego degradation caused by embedding is noticeable. For a hiding ratio of 25% Ranjithkumar et al.[14], achieve a PSNR of 49 dB which is less than Kar et al.[16], Narayanan et al.[19], Balu et al.[20] and Mstafa et al.[22]. In all of these methods discussed above there is no adaptability with respect to the Secret data to be hidden. Choosing an optimal payload is critical in developing an efficient steganographic technique. A three level secured adaptive video steganography technique proposed in this paper resulted a PSNR of 50.2553 dB to 62.0528 dB with hiding ratio of 7.1 to 28.125% (or) payload of 0.56–2. 25 bpp.

The proposed system has the following advantages over the similar works:

- It offers three levels of security when it comes to breaking the information carried secretly.
- It has a combined chaotic system that maintains randomness across the entire range of control parameters.
- The number of secret information bits embedded in each frame of the video cover can be controlled by the user.

The rest of the article is organized as follows. "Chaotic maps" Section describes the various types of chaotic maps that are employed in the scheme. The proposed approach is discussed in "Proposed method" Section, "Performance Analysis" Section provides an overview of performance analysis, while "Conclusion" Section concludes.

Chaotic maps

The logistic map, sine map, and tent map are the most well-known chaotic systems that researchers use to generate random sequence numbers. Maps are expressed mathematically as[12,27]

$$X_{n+1} = bX_n(1 - X_n)\ \left(\text{Logistic Map}\right) \tag{1}$$

$$X_{n+1} = b\sin(\pi_n)\ \left(\text{Sine Map}\right) \tag{2}$$

$$X_{n+1} = \begin{cases} bX_n for X_n < 0.5 \\ b(1 - X_n) for X_n \geqslant 0.5 \end{cases} \left(\text{Tent Map}\right) \tag{3}$$

Table 1 provides competent techniques that are utilizing chaotic dynamic systems for securing information. A single chaotic map[13,16,19] is used to protect the secret data to be transmitted. AES[20,21] and a hybrid of two 1D maps[12,14] is used for encryption and. The proposed method employs a mixed chaotic system to both protect and hide the protected data at random locations. This improves the steganography's quality even further.

Chaotic key generator

The one-dimensional chaotic maps like the logistic and sine map exhibit chaotic and non-chaotic behaviour based on the bifurcation parameter (b)[32]. In Fig. 1a, dark areas (b = 3.57 to 4) indicate chaotic behaviour, while solid areas (b = 0 to 3.57) represent non-chaotic behaviour. However, chaotic output is limited to a small range (0,4). Similar behaviour is observed in the sine map Fig. 1b. The proposed chaotic key generator combines logistic and sine maps, as shown in Fig. 1c. This fusion enables good chaotic behaviour across the entire bifurcation parameter range (0 < b < 1). This ensures sensitivity to initial conditions and randomness throughout the phase space (0 to 1) for secure key generation.

Lyapunov exponent (LE)

The Lyapunov exponent (LE) is used to evaluate the behavior of any discrete time system[11]. The LE is expressed as

$$\lambda(x_0) = n \xrightarrow{Lim} \infty \frac{1}{n}\sum_{i=1}^{\infty} \ln\left|f'(x)\right| \tag{4}$$

where x_0 is the initial value of the map, f' (x) is the derivative of the first order differential equation, and n is the length of the sequence. According to Fig. 2a, b, the logistic map and sine map has chaotic behavior for control parameters ranging from 3.57 to 4 and 0.97 to 1. The proposed CKG- LE Fig. 2c has chaotic behavior over the entire range of control parameter (b). The coupled chaotic system is detailed below.

Figure 3 depicts proposed chaotic key generator (CKG). From the Fig. 3,

References	Map	Level of security for embedding
Younus et al.[13],	Vigenere Cipher $e_k(p_i) = (p_i + k_{(i \bmod m)}) \bmod l$	Two level security
Narayanan[19], et al	3-D Logistic Map $x_{i+1} = \gamma(1 - x_i) + \beta(y_i^2 x_i) + \alpha z_i^3$ $y_{i+1} = \gamma(1 - y_i) + \beta(z_i^2 x_i) + \alpha x_i^3$ $z_{i+1} = \gamma(1 - z_i) + \beta(x_i^2 y_i) + \alpha y_i^3$	Single level security
Kar[16], et al	Burger 2-D chaotic map $X_{n+1} = X_n^2 - Y_n^2 + aX_n + bY_n$ $Y_{n+1} = 2X_nY_n - Y_n^2 + cX_n + dY_n$	Two level security
Balu[20], et al	Advanced Encryption Standard	Two level security
Abed et al.,[21]	Advanced Encryption Standard	Two level security
Ranjithkumar[12], et al	Logistic Map and Tent map $X_{n+1} = bX_n(1 - X_n)X_{n+1} = \begin{cases} \mu X_n for X_n < 0.5 \\ \mu(1 - X_n) for X_n \geqslant 0.5 \end{cases}$	Three level security
Ranjithkumar[14], et al	Logistic Map $X_{n+1} = bX_n(1 - X_n)$ and Tent map $X_{n+1} = \begin{cases} \mu X_n for X_n < 0.5 \\ \mu(1 - X_n) for X_n \geqslant 0.5 \end{cases}$	Three level security
Proposed method	$CKG = mod\ (4\ \sin\theta_1\ 1 - \sin(\theta_1)) + \sin(\theta_2), 2$	Three level security

Table 1. Chaotic maps and level of Security by similar works.

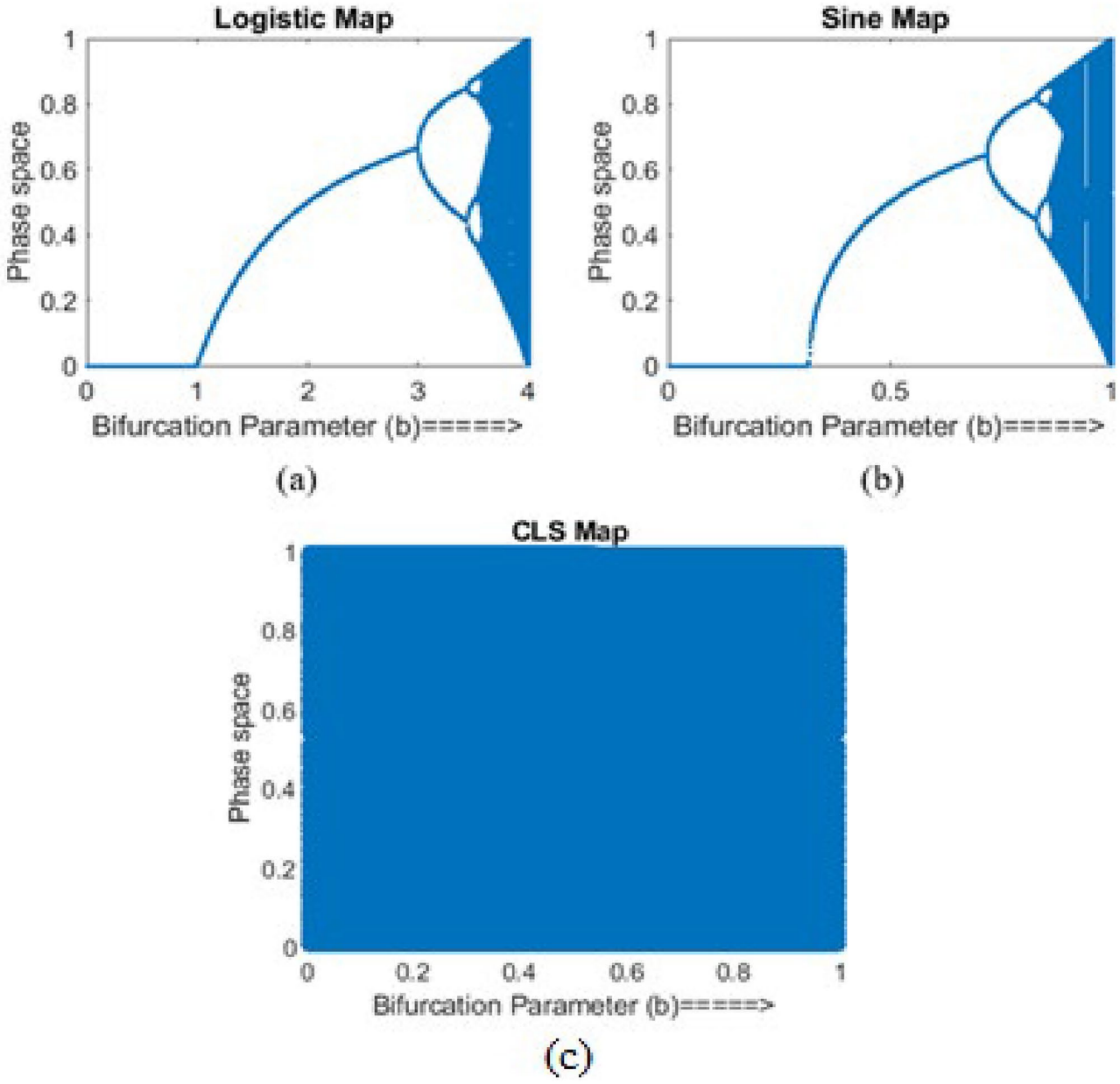

Figure 1. Bifurcation diagram (**a**) logistic map (**b**) sine map (**c**) combined logistic-sine map.

$$IP_1 = b_{L2} b_{s1} [\sin(\pi_{ns1}) - b_{s1} sin^2(\pi_{ns1})] \tag{5}$$

$$IP_2 = X_{nS2+1=} b_{s2} \sin(b_{L1} X_{nL1}(1 - X_{nL1})) \tag{6}$$

$$IP_3 = bitXor[5 + 6] \tag{7}$$

Substituting $b_{L2} \cong 4$, $b_{s1} \cong 1, b_{s2} \cong 1$

$$IP_3 = bitXor\{4 \ \sin(\pi_{ns1}) - 4sin^2(\pi_{ns1}) + \ \sin(\pi_{nL1})\}$$

IP_3 always lies between (0,1)

$$\pi_{nS1} \ \rightarrow \ \theta_1, \ \pi_{nS2} \ \rightarrow \ \theta_2$$

$$CKG = mod(4 \sin\theta_1(1 - \sin(\theta_1) + \sin(\theta_2), 2) \tag{8}$$

The key generator block (KGB) generates internal keys[32] from external keys provided by authorized users[32]. These internal keys are utilized as kernels for frame selection (FS), permutation order (PO), and random position

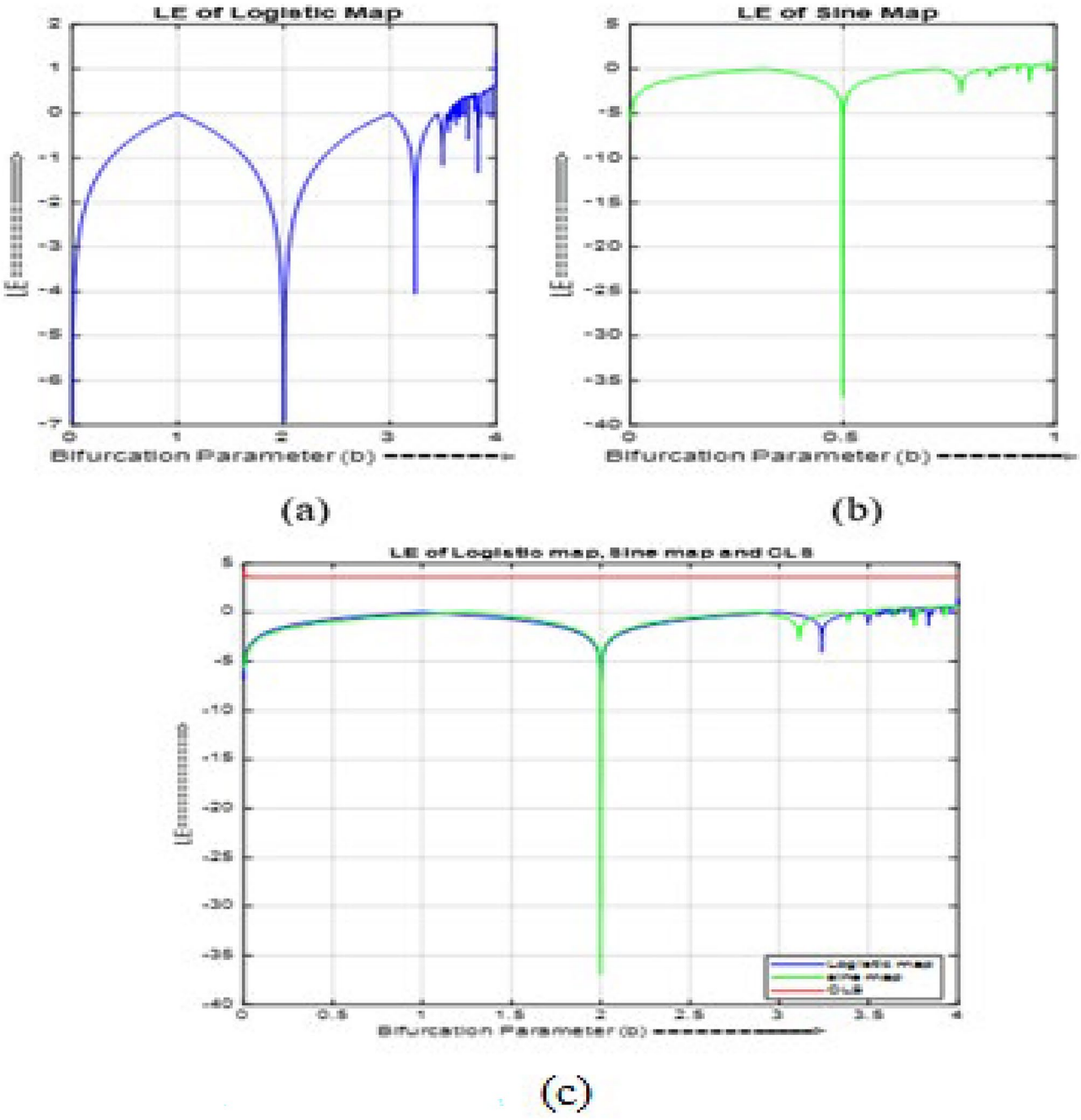

Figure 2. Lyapunov exponent of (**a**) logistic map (**b**) sine map (**c**) Combined logistic-sine map.

hiding (RPH). The KGB enhances sensitivity to the extent that a single bit change in any external key causes a significant alteration in the internal keys. This increased sensitivity is achieved through the use of chaotic maps.

Proposed method

Figure 4 depicts a diagram of the proposed adaptive video steganography algorithm. Table 2 shows the pseudocode that explains the proposed algorithm. In pseudocode, the prefix '##' is used to denote each block of Fig. 4. The process of each block is explained below.

Algorithm

Step 1 Get input cover video (C_{VF}) and secret data (S_D) in input block (## input block). Compute the dimensions (size_C_{VF}, size_S_D, FN) of them and verify the possibility of embedding using ACB block.

Step 2 ACB block throws the possibility of embedding (pos) and number of bits to be embedded (bits_per_frame) into each frame (bits_per_frame). (refer ## ACB block).

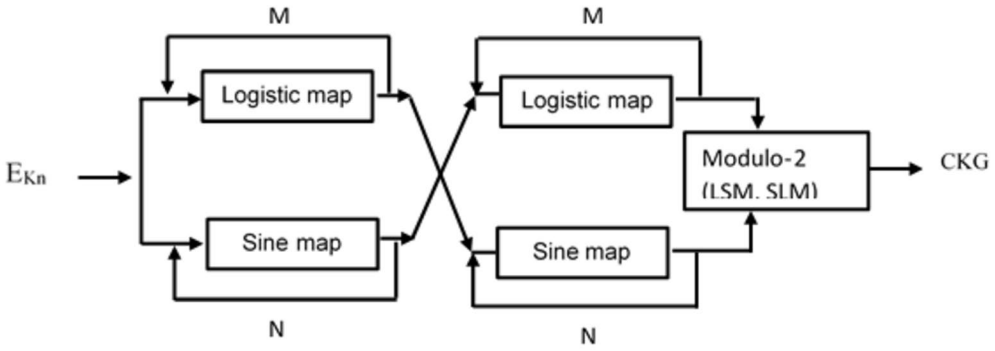

Figure 3. Proposed chaotic key generator (CKG).

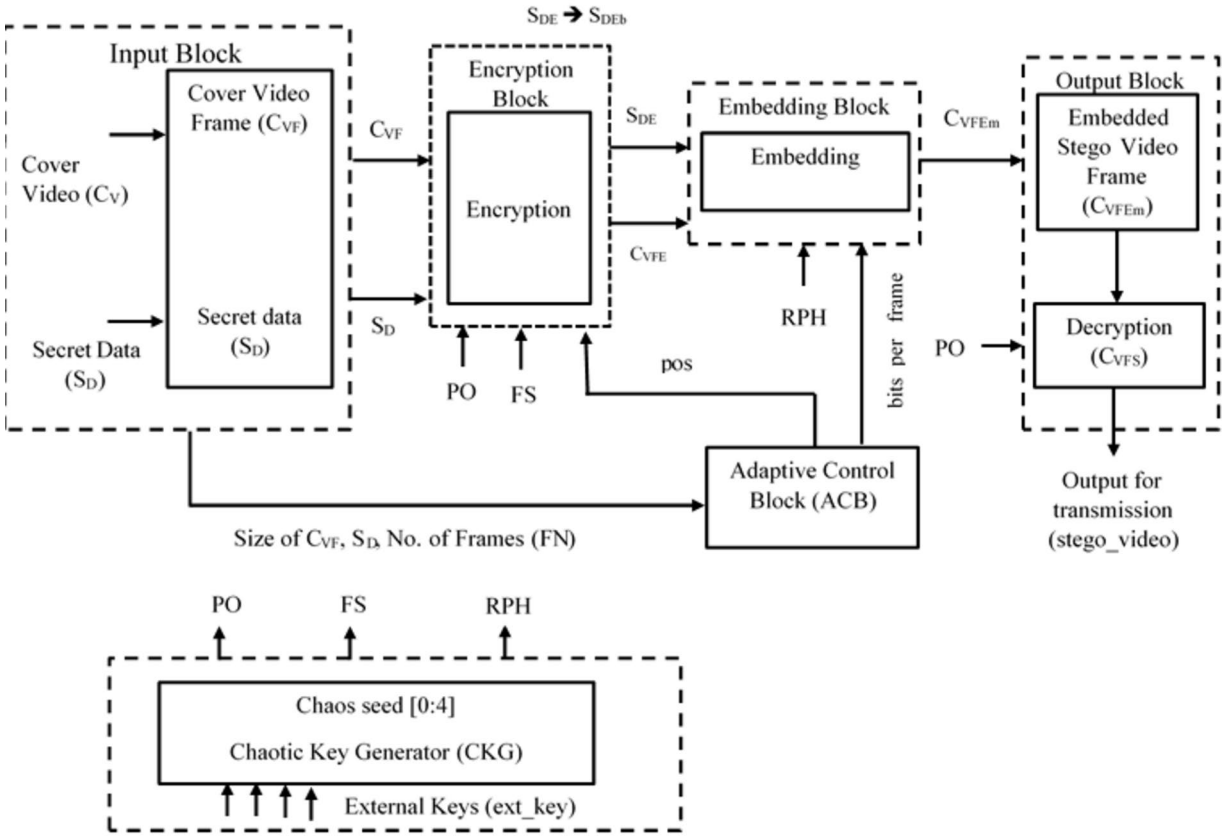

Figure 4. Block diagram proposed adaptive video steganography method.

Step 3 If pos from ACB is True: Generate the keys (chaos_seed) from user input(ext_key) and compute the parameters (PO_1, PO_2, FS and RPH) using key generation block (##key_generation). These are required to encrypt S_D and C_{VF}. if pos from ACB is False: jump to Step 7.

Step 4 Encrypt the C_{VF} using PO_2 and S_D using PO_1 at Encryption block(##Encryption). store the results in C_{VFE} and S_{DE}. Permutation is carried out in encryption.

Step 5 Convert S_{DE} into binary S_{DEb} and slice it based on the bits_per_frame variable. The sliced S_{Deb}, C_{VFE}, bits_per_frame and RPH are then transferred to embedding.

Step 6 All embedded frames (C_{VFEm}) obtained in step 5 are permuted ones. Apply reverse permutation to all using decryption block (##Decryption). The output of the block is C_{VFS}. Combine all C_{VFS} to get Stego video frame (Stego_video).

Step 7 exit ().

```
## main program
## 1 input block (step1)
# CV → input cover video,  SD → secret data
# convert video into frames
CVF ← extract_frames (CV, frames_per_sec)
FN ← len (CVF)     # Number of frames in CVF
size_CVF , size_SD ← CVF.size(),SD.size()
(step2)
# L → depth of pixel in bits, HR → hiding ratio
[pos, bits_per_frame] = ACB(size_CVF, size-SD, FN,
L=8, HR = 28.125)
if pos == True: (Step 3)
   #ext_key → 212 bit random bit stream
   ext_key ← input(get_data_bit_stream_from_user)
   # chaos_seed → 4 initial seeds in the range of [0,1]
   chaos_seed ← chaos_key(ext_key)
  # generate PO, FS and RPH
  [PO1,PO2] ← PO_gen (size_CVF, size_SD,
chaos_seed)
  FS ← FS_gen (FN, chaos_seed)
  RPH ← RPH_gen(size(CVF),chaos seed)
   RPH_slice ← slice RPH into 3 , each of size equal to
size_CVF
  ## transfer to encryption block (step4)
  for i in range(FN):
      CVFE[i] ← permute_CVF(CVF[i],PO2)
   SDE ← permute_secretdata(SD,PO1)
   ## Embedding (step5)
   SDEb ← bin(SDE)        #binary conversion
   SDEB_sliced ← slice SDEb into FN pieces of length…
                  … bits_per_frame
   for i in range(FN):
      CVFEm[i] ← embedding (CVFE
[FS[i]],SDEb_sliced[i], RPH_slice, Bits_per_frame)
```

```
## ACB block (step2)
def ACB(size_CVF,size_SD, FN,L,HR):
   EC_per_frame = size_CVF * L* HR
   SD_size_in_bits = size_SD * 8
   if (SD_size_in_bits < EC_per_frame):
      pos = True
      bits_per_frame = SD_size_in_bits…. …
//EC_per_frame
   else:
     pos = False
    bits_per_frame = 0
   return pos, bits_per_frame
##key_generation (step3)
def chaos_key(ext_key):
   return chaos_seed = [apply equation 9 to
ext_key]
def logistic(seed,M=1):
   return last value of equation 1 after M
iterations
def sine(seed,N=1):
   return last value of equation 2 after N
iterations
def tent(seed,N=1):
   return last value of equation 3 after N
iterations
def combined_chaotic(seed):
   ip1 ← sine(logistic(seed[0],1500),1500)
   ip2 ← logistic(sine(seed[1],1500),1500)
   return PP_seed ← bitxor(ip1,ip2)
def PO_gen (size_CVF, size_SD, chaos_seed):
    PO_seed[0] ←
combined_chaotic(chaos_seed[0])
    PO_seed[1] ←
combined_chaotic(chaos_seed[1])
```

(continued)

Performance analysis

The effectiveness of the proposed adaptive video steganographic method is evaluated by three key measures namely Imperceptibility, hiding ratio (HR) and payload(P)[4,5,25]. The proposed method has been tested on several video sequences downloaded from[19] using Phycharm IDE with python 3.7, windows 10, intel(R) Core ™ i3 processor @ 2.4 GHz, with 4 GB RAM and secret data to be embedded in the cover video frame has been downloaded from USC-SPIC image database[29] and ITU-T test signal for telecommunications systems[30]. The performance analyses are as follows.

Imperceptibility

Peak signal-to-noise ratio (PSNR)[4,5,16] is used to assess the imperceptibility of the proposed method. The PSNR is calculated using Eq. (10) and measured in decibels[4,5,16].

$$PSNR = 10\log_{10}\frac{(2^L)^2}{MSE(C_{VF}, C_{VFS})} \tag{10}$$

where L is the image's depth and MSE is the mean square error, which is calculated as

$$MSE = \frac{1}{MxN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(C_{VF,}C_{VFS})^2 \tag{11}$$

The ideal PSNR value is greater than 30 dB[22,31]. The stego degradation caused by embedding is noticeable if the PSNR value is less than 30 dB[22,31]. Figures 5 and 6 show the original video frames as well as the stego video frames. From Figs. 5 and 6, the difference between the stego video. frame and the original video frame is indistinguishable.

PSNR values of different video sequences are embedded with 25%, 50%, 75% and 100% embedding capacity of C_{VF} using images and audio waves as secret data is shown in Table 3. The results show that the proposed method PSNR value is 50 to 62 dB, which is greater than the ideal value of 30 dB[22,31], implying that the stego video is similar to the original cover video and has greater security than other methods.

Table 4 shows comparison of PSNR for various embedding capacities using image and audio as secret data. PSNR below 30 dB suggests that the embedded data has caused perceptible changes in the video, potentially compromising the secrecy of the hidden information and reducing the overall effectiveness of the steganography

```
#Decrytion to get stego frames (step6)
for i in range(FN):
    C_VFS[i] ← decrypt(C_VFEm[i], PO_2)

Stego_video ← combine all C_VFS
else: (step7)
print('Embedding isn't possible !!! Secret data size is greater than embedding capacity')
##return to main
```

```
for i in range(size_SD):
    PO_seed[0] ← logistic(PO_seed[0])
    PO_1[i] ← floor( (PO_seed[0]*10^14) % size_S_D)
for j in range(size_CVF):
    PO_seed[1] ← logistic(PO_seed[1])
    PO_2[i] ← floor( (PO_seed[1]*10^14) % size_C_VF)
return PO_1, PO_2
def FS_gen(FN, chaos_seed):
    FS_seed ← combined_chaotic(chaos_seed[2])
for i in range(l):
    FS_seed ← sine(FS_seed)
    FS[i] ← floor( (FS_seed*10^14) % FN)
return FS
#MNP is size of a frame (C_VF)
def RPH_gen(MNP,chaos_seed):
    RPH_seed ← combined_chaotic(chaos_seed[3])
for i in range(MNP*3):
    RPH_seed ← tent(RPH_seed)
    RPH[i] ← floor((RPH_seed * 10^14) % MNP)
return RPH
```

```
Notations:
C_V ← Cover video of size M ×N ×P ×frame F_N
S_D ← Secret data of max size 28.125 % of C_V
C_VF ← cover video frames (FN numbers)
FN ← total number of frames in C_VF
Ext_key ← given by the user (212 bits)
Chaos-seed ← internal keys generated from ext_key (4 Numbers)
PO_1 ← permutation order of size equal to S_D
PO2 ← permutation order of size equal to C_VF
FS ← frame selection of size equal to FN
RPH ← random postions for hiding secret information bits into LSB, 1st ISB and 2nd ISB (size 3 x M× N× P)
C_VFE ← Encrypted cover video frames
S_DE ← Encrypted secret data
S_DEb ← secret data in binary format
C_VFEm ← embedded C_VFE frames
C_VFS ← stego video frames
```

```
## Encryption (step4)
# C_VF → cover video frame of size M×N×P row # vector
# PO_2 → permutation order of size M×N×P
def permute_C_VF (C_VF, PO_2)
for i in range (M×N×P):
    C_VFSE[i] ← C_VF[PO_2[i]]
return C_VFSE
# S_D → secrtet data of size m×n
# PO_1 → permutation order of size m×n
def permute_secretdata(S_D,PO_1):
for i in range (len(S_D)):
    S_DE [i] ← S_D[PO_1[i]]
return S_DE
```

```
## Embedding (step5)
# in bit plane 7 stands for LSB
def embedding (C_VF, S_D, RPH_slice, Bits_per_frame):
C_VF_Bit_plane[0:7] ← extract bit planes of C_VF
if Bits_per_frame <= len(RPH_slice[0]):
    C_VF_bit_plane[7] ← S_D[RPH_slice[0]]
elif Bits_per_frame > len(RPH_slice[0]) <= len(2*RPH_slice[0]):
    C_VF_bit_plane[7] ← S_D[RPH_slice[0]]
    C_VF_bit_plane[6] ← S_D[RPH_slice[1]]
else:
    C_VF_bit_plane[7] ← SD[RPH_slice[0]]
    C_VF_bit_plane[6] ← SD[RPH_slice[1]]
    C_VF_bit_plane[5] ← SD[RPH_slice[2]]
C_VFE ← combine bit planes and convert back to integer
return C_VFE
```

```
## Decryption (step6)
# C_VFE → Encrypted and embedded video frame of #size M×N×P row vector
# PO_2 → permutation order of size M×N×P
def repermute_CVF (C_VF, PO_2)
for i in range (M×N×P):
    C_VFS[PO_2[i]] ← C_VFE[i]
return C_VFS
```

Table 2. Pseudocode for the proposed video steganography method.

Figure 5. Original video frames of mobile calendar (Frame Number: 75,152 and 251).

Figure 6. Stego video frames of mobile calendar (Frame Number: 75,152 and 251).

method. Therefore, maintaining a PSNR above 30 dB is crucial for ensuring both effective data concealment and visual integrity in steganographic applications.

Hiding ratio(HR)

The hiding ratio is a measurement of the percentage of space in the cover frame that can be used to embed secret data[4,5,20]. The following equation is used to calculate the hiding ratio.

$$HR = \frac{Sizeof\ \sec retdata}{Sizeof\ \mathrm{cov}\ ervideo} X100 \tag{12}$$

The proposed method embeds the secret data using LSB, 1st ISB, and 25% of the 2nd ISB of the cover frame. As a result, the hiding ratio ranges from 7.1 to 28.125%. Table 5 compares the proposed method to other methods, and Fig. 7 show comparison in graphical format of HR, Payload and PSNR. Table 5 shows that the proposed method outperforms other methods in terms of PSNR, HR, and payload.

Payload

The payload represents the maximum number of bits allocated for embedding the secret data within the cover frame, which is calculated in terms of bits per pixel using the following equation (bpp)[5,14].

$$P = \frac{\|S\|}{MXN} \tag{13}$$

where $\|S\|$ denotes the number of secret bits to be embedded in the cover frame. M and N are the cover frames' height and width. To embed the secret data, the proposed method uses 28.125 percent of the cover data. As a result, the proposed method's payload is

$$P = \frac{HidingRatio}{100} X8 \tag{14}$$

$$P = \frac{28.125}{100} X8 = 2.25bpp \tag{15}$$

The adaptive control block (ACB) allows users to choose between partial and full embedding capacities in the cover video frames based on the amount of secret data to be embedded. Specifically, the proposed method uses 28.125% of the cover data, resulting in a payload of 2.25 bits per pixel (bpp). If the number of bits required to embed the secret data is less than the payload, the ACB enables the user to select partial embedding capacity. If the number of bits exceeds the payload, the full embedding capacity is selected. Table 5 compares the proposed method to other methods in terms of payload. The proposed method's payload is observed to be greater than that of other methods.

S. no	Video file name	Embedding capacity	Images				Audio waves			
			Lena	Peppers	Mandrill	Splash	Female1. wav	Female2. wav	Male1. wav	Male2. wav
			PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR
1	Bus	100	50.2739	50.2745	50.2736	50.2776	50.0068	50.0909	50.1228	49.9860
		75	52.0396	52.0398	52.0360	52.0412	51.8750	51.9436	51.9679	51.8642
		50	55.0543	55.0414	55.0535	55.0478	54.8988	54.9418	54.9751	54.8931
		25	62.0424	62.0634	62.0431	62.0456	59.0239	59.0316	59.0197	58.9889
2	Carphone	100	50.2826	50.2828	50.2834	50.2833	50.0360	50.1202	50.1386	50.0034
		75	52.0453	52.0466	52.0424	52.0476	51.8825	51.9528	51.9799	51.8410
		50	55.0527	55.0529	55.0565	55.0552	54.9328	54.9696	54.9776	54.9243
		25	62.0445	62.0496	62.0472	62.0439	58.9847	58.9619	58.9612	58.9336
3	City	100	50.2802	50.2849	50.2789	50.2820	50.0514	50.1032	50.1238	50.0082
		75	52.0394	52.0411	52.0445	52.0416	51.9019	51.9687	51.9731	51.8694
		50	55.0536	55.0521	55.0594	55.0558	55.0101	55.0386	55.0580	54.9887
		25	62.0528	62.0422	62.0442	62.0426	58.9820	59.0036	58.9834	58.9834
4	Container	100	50.2553	50.2560	50.2574	50.2553	49.8265	49.9290	49.9521	49.8136
		75	52.0234	52.0239	52.0272	52.0286	51.6884	51.7804	51.8102	51.6770
		50	55.0412	55.0426	55.0449	55.0413	54.7796	54.8381	54.8459	54.7473
		25	62.0418	62.0481	62.0400	62.0468	58.8277	58.8324	58.8546	58.7731
5	Foreman	100	50.2737	50.2718	50.2694	50.2553	49.9160	50.0015	50.0415	49.8965
		75	52.0374	52.0375	52.0354	52.0286	51.7313	51.8099	51.8627	51.7272
		50	55.0448	55.0474	55.0448	55.0413	54.8427	54.8814	54.8893	54.8301
		25	62.0440	62.0410	62.0520	62.0436	58.9254	58.9536	58.9764	58.9035
6	Husky	100	50.2777	50.2786	50.2782	50.2790	50.0156	50.1004	50.1591	49.9999
		75	52.0425	52.0400	52.0392	52.0393	51.8479	51.9218	51.9628	51.8375
		50	55.0555	55.0520	55.0501	55.0563	54.9015	54.9473	54.9608	54.8871
		25	62.0445	62.0433	62.0370	62.0364	58.9467	58.9432	59.0036	58.9425
7	Miss	100	50.2704	50.2750	50.2768	50.2715	49.9605	50.0128	50.0364	49.9361
		75	52.0348	52.0389	52.0385	52.0382	51.7910	51.8384	51.8707	51.7815
		50	55.0489	55.0567	55.0540	55.0524	54.8558	54.8768	54.9172	54.8432
		25	62.0294	62.0441	62.0438	62.0379	58.8526	58.8844	58.8688	58.8405
8	Mobile calendar	100	50.2781	50.2775	50.2761	50.2778	50.0455	50.1190	50.1491	50.0581
		75	52.0379	52.0406	52.0409	52.0394	51.8946	51.9302	51.9535	51.9011
		50	55.0532	55.0533	55.0532	55.0550	54.9915	54.9920	54.9818	54.9680
		25	62.0449	62.0437	62.0350	62.0441	58.9785	59.0057	58.9799	58.9980
9	News	100	50.2807	50.2808	50.2812	50.2802	49.9776	50.0335	50.0849	49.9442
		75	52.0461	52.0487	52.0469	52.0450	51.8173	51.8931	51.9113	51.8127
		50	55.0554	55.0560	55.0537	55.0543	54.8462	54.8752	54.8609	54.8453
		25	62.0443	62.0436	62.0457	62.0448	58.9151	58.9212	58.8688	58.8824
10	Soccer	100	50.2820	50.2827	50.2823	50.2813	50.0612	50.1421	50.1745	50.0449
		75	52.0444	52.0438	52.0470	52.0447	51.9303	51.9657	51.9729	51.9196
		50	55.0500	55.0524	55.0491	55.0517	54.9619	54.9967	54.9965	54.9512
		25	62.0526	62.0441	62.0507	62.0522	58.9481	58.9584	58.9924	58.9674
Min–Max			50.255-62.0528	50.2808-62.0496	50.2574-62.052	50.280-62.0522	49.8265-59.0239	50.0015-59.0057	49.9521-59.0197	49.8136-58.9980

Table 3. PSNR of partial (or) full embedding capacity of cover frame with secret data as images and audio waves.

Embedding capacity	PSNR (dB)	
	Video	Audio
25%	62.05	59
50%	55.06	54.74
75%	52.05	51.67
100%	50.25	49.81

Table 4. Comparison of PSNR for various embedding capacity using image and Audio as secret data.

Method	Payload (bpp)	Hiding ratio (%)	PSNR (dB)
Kacar et al.[4]	0.6	8.08	55
Darani et al.[5]	0.22	2.77	47.8
Zakaria et al.[7]	3.24	40.5	42.15
Setiadi[8]	1.25	15.62	44
Abd-El-Atty[10]	2	25	44.1
Taha et al.[9]	1.125	14.06	55.47
Ranjithkumar et al.[5]	2	25	49
Younus et al[4]	1.59	–	55.71
Balu et al[11]	0.1	–	67.12
Kar et al[7]	1.78	–	52.24
Mstafa et al[13]	–	3.46	49.01
Alavianmehr et al[16]	1.34	–	36.97
Hu et al[17]	4	–	29.75
Proposed method	2.25	28.12	50.25 dB to 62.05 dB

Table 5. Comparison of proposed method with similar methods.

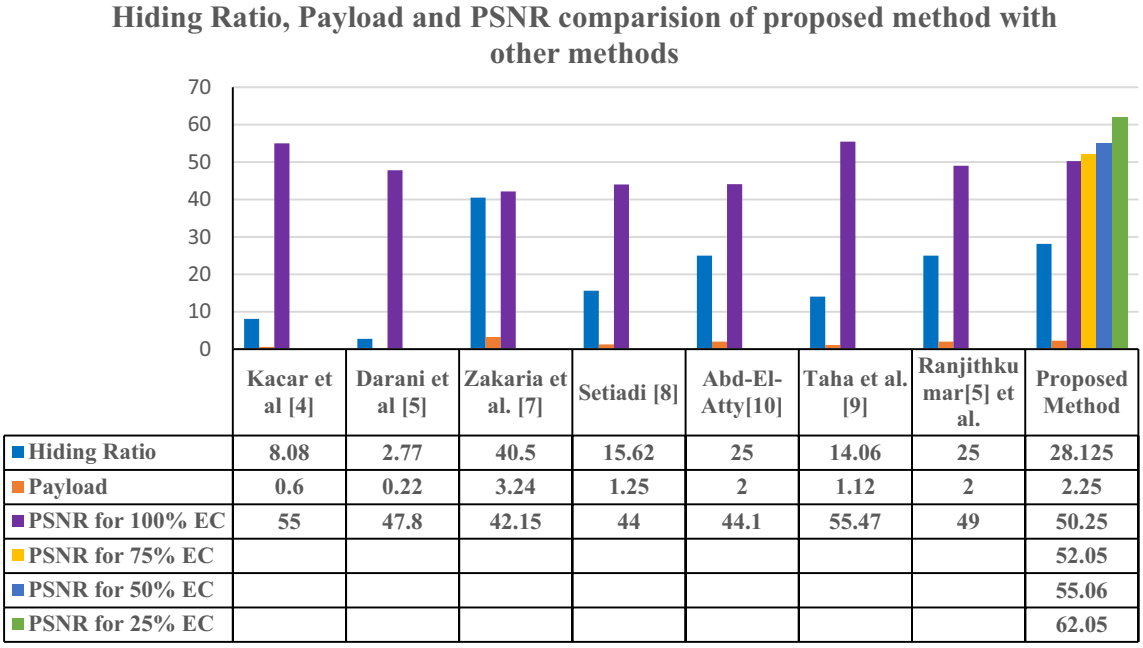

Figure 7. Hiding Ratio, Payload and PSNR Comparison.

Randomness test with SP800-22 test suite

The NIST test is one of the most important authorized standards for determining the randomness of image obtained using the suggested algorithm[32]. To demonstrate the randomness of the binary sequence generated by the proposed approach, we employ 16 statistical tests from the NIST test suite. The evaluation is based on the binary sequence's *P*-values in each test. If the *P*-values for each test are ≥ 0.01, it means the created binary sequence is random and evenly distributed. If the *P*-values are less than 0.01, the generated sequence is not random and has an uneven distribution. To confirm the uniform distribution of *P*-values, we evaluate the distribution for a large number of binary sequences (N = 100) for each test. The computation is as follows:

$$\chi^2 = \sum_{i=1}^{10} \left(\frac{F_i - N/10}{N/10} \right) \tag{16}$$

where Fi is the number of occurrences the *P*-value in the ith interval and N denotes the sample size (N = 100). The *P*-value of *P* values are calculated by using the following formula

Statistical test		Diffusion	
		P-value	Result
Frequency		0.1831	Pass
Block frequency		0.4380	Pass
Runs		0.7519	Pass
Statistical test		0.4803	Pass
Long runs of one's		1.0000	Pass
Binary matrix RANK		0.7522	Pass
DFT		1.0000	Pass
Non overlapping Templates		0.5598	Pass
Overlapping templates		0.9990	Pass
Universal		0.9985	Pass
Serial	*P* value 1	0.7550	Pass
	P value 2	0.3683	Pass
Approximate entropy		0.2049	Pass
Cumulative sums		0.6383	Pass
Random excursion		0.1301	Pass
Random excursion variant		0.6394	Pass
Final result		Pass	

Table 6. NIST Statistical test results for encrypted image using proposed method.

$$Pvalue = igamc\left(\frac{9}{2}, \frac{\chi^2}{2}\right) \tag{17}$$

where 'igamc' is the incomplete Gamma function. The results of each statistical test are shown in Table 6. The results show that the embedded image has passed all tests and distribution is uniform. NIST test results prove that the embedded image is random.

Conclusion

This paper proposes a three-layer secured adaptive video steganography based on chaotic systems. The transmitted information is hidden within the video frames' (C_{VF}) in the spatial domain. The method allows the user to select either partial embedding capacity of the C_{VF} or full embedding capacity of the C_{VF}. Secret information to be transmitted is encrypted and hidden at random positions within encrypted C_{VF}. Permutation of video frames (C_{VF}) and secret information by PO offers the first layer of security, selection of C_{VFs} through FS for embedding provides second layer of security and third layer of security is provided by the RPH, the positions for hiding information bits randomly. CKG is responsible for generating PO, FS and RPH. CKG structure consists of one-dimensional logistic and Sine maps, which are aligned in such a way that they increase the sensitivity of key production, enhance the randomness and thus improve the quality of the embedding procedure. The method's competence is demonstrated through evaluation methodologies such as PSNR, HR, and payload. It provides a maximum payload of 2.25 bpp, hiding ratio of 7.1% to 28.125% and PSNR of 50.25 to 62.05 dB. The evaluation results and comparisons with the similar methods show that the method proposed is better than other methods. Our future work is to generate unbreakable key structure using machine learning and implement for video steganography.

Data availability

The datasets used and analyzed during the current study are available from the corresponding author on request.

Received: 27 February 2024; Accepted: 8 July 2024
Published online: 07 August 2024

References:

1. Khan, A. & Sarfaraz, A. Vetting the security of mobile applications. *Sci. Int.* **29**(2), 361–365 (2017).
2. Khan, A. Comparative analysis of watermarking techniques. *Sci. Int.* **27**(6), 6091–6096 (2015).
3. Stallings, W. *Cryptography and network security: principles and practice* 3rd edn. (PHI, Delhi, 2006).
4. Kacar, S. *et al.* 4D chaotic system-based secure data hiding method to improve robustness and embedding capacity of videos. *J. Inf. Secur. Appl.* **71**, 103369. https://doi.org/10.1016/j.jisa.2022.103369 (2022).
5. Darani, A. Y. *et al.* Optimal location using genetic algorithms for chaotic image steganography technique based on discrete framelet transform. *Digit. Signal Process.* **144**, 104228. ISSN 1051-2004. https://doi.org/10.1016/j.dsp.2023.104228 (2024).
6. Aparna, H. & Madhumitha, J. Combined image encryption and steganography technique for enhanced security using multiple chaotic maps. *Comput. Electric. Eng.* **110**, 108824. ISSN:0045-7906. https://doi.org/10.1016/j.compeleceng.2023.108824 (2023).
7. Zakaria, A. A. *et al.* High-capacity image steganography with minimum modified bits based on data mapping and LSB substitution. *Appl. Sci.* **8**, 2199. https://doi.org/10.3390/app8112199 (2018).

8. Setiadi, D. R. I. M. Improved payload capacity in LSB image steganography uses dilated hybrid edge detection. *J. King Saud Univ. Comput. Inf. Sci.* **34**(2), 104–114. https://doi.org/10.1016/j.jksuci.2019.12.007 (2022).
9. Taha, M. S. *et al.* High payload image steganography scheme with minimum distortion based on distinction grade value method. *Multimed. Tools Appl.* **81**, 25913–25946. https://doi.org/10.1007/s11042-022-12691-9 (2022).
10. Abd-El-Atty, B. A robust medical image steganography approach based on particle swarm optimization algorithm and quantum walks. *Neural Comput. Appl.* **35**, 773–785. https://doi.org/10.1007/s00521-022-07830-0 (2023).
11. Hua, Z., Zhou, B. & Zhou, Y. Sine-transform-based chaotic system with FPGA implementation. *IEEE Trans. Ind. Electron.* **65**(3), 2557–2566. https://doi.org/10.1109/TIE.2017.2736515 (2018).
12. Ranjith Kumar, R., Jayasudha, S. & Pradeep, S. Efficient and secure data hiding in encrypted images: a new approach using chaos. *Inf. Secur. J.* **25**(4–6), 235–246. https://doi.org/10.1080/19393555.2016.1248582 (2016).
13. Younus, Z. S. & Hussain, M. K. Image steganography using exploiting modification direction for compressed encrypted data. *J. King Saud Univ.-Comput. Inf. Sci.* **34**(6), 2951–2963. https://doi.org/10.1016/j.jksuci.2019.04.008 (2022).
14. Ranjithkumar, R., Ganeshkumar, D. & Senthamilarasu, S. Efficient and secure data hiding in video sequence with three-layer security: an approach using chaos. *Multimed. Tools Appl.* **80**(9), 13865–13878. https://doi.org/10.1007/s11042-020-10324-7 (2021).
15. Tao, H., Chongmin, L., Jasni, M. Z. & Abdalla, A. N. Robust image watermarking theories and techniques: a review. *J. Appl. Res. Technol.* **12**(1), 122–138. https://doi.org/10.1016/S1665-6423(14)71612-8 (2014).
16. Kar, N., Mandal, K. & Bhattacharya, B. Improved chaos-based video steganography using DNA alphabets. *ICT Express* **4**(1), 6–13. https://doi.org/10.1016/j.icte.2018.01.003 (2018).
17. Abdulla, A. A., Sellahewa, H. & Jassim, S. A. Improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images. *Multimed. Tools Appl.* **78**(13), 17799–17823. https://doi.org/10.1007/s11042-019-7166-7 (2019).
18. Zhang, L. & Chen, D. The large capacity embedding algorithm for H.264/AVC intra prediction mode video steganography based on linear block code over Z4. *Multimed. Tools Appl.* **79**(17–18), 12659–12677. https://doi.org/10.1007/s11042-019-08528-7 (2020).
19. Narayanan, G., Narayanan, R., Haneef, N., Chittaragi, N. B. & Kodagudi, S. G. A novel approach to video steganography using 3D chaotic map. In *TENCON 2019 IEEE Region 10 Conference (TENCON)* 955–959. https://doi.org/10.1109/TENCON.2019.8929347
20. Balu, S., Babu, C. N. K. & Amudha, K. Secure and efficient data transmission by video steganography in medical imaging system. *Clust. Comput.* **22**(2), 4057–4063. https://doi.org/10.1007/s10586-018-2639-4 (2019).
21. Abed, S. *et al.* An automated security approach of video steganography-based on LSB using FPGA implementation. *J. Circuits Syst. Comput.* **28**(05), 1950083. https://doi.org/10.1142/S021812661950083X (2018).
22. Mstafa, R. J., Elleithy, K. M. & Abdelfattah, E. A robust and secure video steganography method in DWT-DCT domains based on multiple object tracking and ECC. *IEEE Access* **5**, 5354–5365. https://doi.org/10.1109/ACCESS.2017.2691581 (2017).
23. Mstafa, R. J. & Elleithy, K. M. A novel video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes. In *2015 long island systems, applications and technology*, 1–7. https://doi.org/10.1109/LISAT.2015.7160192.
24. Kelash, H.M., Abdel Wahab, O.F., Elshakankiry, O.A. & El-sayed, H.S. Hiding data in video sequences using steganography algorithms. In *2013 International Conference on ICT Convergence (ICTC)* 353–358. https://doi.org/10.1109/ICTC.2013.6675372 (2013).
25. Alavianmehr, A., Rezaei, M., Helfroush, M. S. & Tashk, A. A lossless data hiding scheme on video raw data robust against H.264/AVC compression. In *2012 2nd International eConference on Computer and Knowledge Engineering (ICCKE)* 194–198. https://doi.org/10.1109/ICCKE.2012.6395377 (2012).
26. Hu, S.D. & Tak, K.U. A novel video steganography based on non-uniform rectangular partition. In *2011 14th IEEE International Conference on Computational Science and Engineering*. 57–61. https://doi.org/10.1109/CSE.2011.24 (2011).
27. Kumar Patro, K. A. & Acharya, B. An efficient color image encryption scheme based on 1-D chaotic maps. *J. Inf. Secur. Appl.* **46**, 23–41. https://doi.org/10.1016/j.jisa.2019.02.006 (2019).
28. Video Dataset: https: //media.xiph.Org/video/derf/y4m.
29. "The USC-SIPI Image Database," http://sipi.usc.edu/database Accessed 12 Jun 2017.
30. Speech dataset: https://www.itu.int/net/itu-t/sigdb/genaudio/AudioForm-g.aspx?val=10000501
31. Chang, C., Kieu, T.D., & Chou, Y., A high payload steganographic scheme based on (7, 4) hamming code for digital images. In *2008 International symposium on electronic commerce and security*. 1pp. 6–21, https://doi.org/10.1109/ISECS.2008.222 (2008)
32. Kumar, D., Sudha, V. K. & Ranjithkumar, R. A one-round medical image encryption algorithm based on a combined chaotic key generator. *Med. Biol. Eng. Comput.* **61**, 205–227. https://doi.org/10.1007/s11517-022-02703-z (2023).
33. Kumar, D. & Sudha, V. K. A Hybrid Image Steganography Method Based on Spectral and Spatial Domain with High Hiding Ratio. In: Abraham, A., Pllana, S., Casalino, G., Ma, K. & Bajaj, A. (eds) *Intelligent systems design and applications*. ISDA 2022. Lecture Notes in Networks and Systems, vol. 715. Springer, Cham. https://doi.org/10.1007/978-3-031-35507-3_7(2023).
34. Gopalakrishnan, T. & Ramakrishnan, S. Image encryption using hyper-chaotic map for permutation and diffusion by multiple hyper-chaotic maps. *Wirel. Pers Commun.* **109**, 437–454. https://doi.org/10.1007/s11277-019-06573-x (2019).

Author contributions

Conceptualization, K.D, S.V.K, N.M, and K.R.; Data curation, K.D, S.V.K, N.M, and K.R; Analysis and validation, K.D, S.V.K, N.M, and K.R; Formal analysis, K.D., S.V.K, N.M, and K.R.; Investigation methodology, K.D, S.V.K, N.M, and K.R.; Project administration, K.R.; Software, K.D, S.V.K, N.M, and K.R., Supervision, K.R.; Validation, K.D, S.V.K, N.M, and K.R.; Visualization, K.D, S.V.K, N.M, and K.R.; Writing—original draft, K.D, S.V.K, N.M, and K.R., data visualization, editing and rewriting, K.D, S.V.K, N.M, and K.R.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.