Received: 6 September 2021 | Revised: 10 November 2021 | Accepted: 11 November 2021

IET Smart Cities

DOI: 10.1049/smc2.12019

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Lightweight frame scrambling mechanisms for end-to-end privacy in edge smart surveillance

Alem Fitwi[1] | Yu Chen[1] | Sencun Zhu[2]

[1]Department of Electrical and Computer Engineering, Binghamton University, Binghamton, New York, USA

[2]Department of Computer Science and Engineering, Penn State University, University Park, Pennsylvania, USA

Correspondence

Yu Chen, Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA.
Email: ychen@binghamton.edu

Abstract

As smart surveillance has become popular in today's smart cities, millions of closed circuit television cameras are ubiquitously deployed that collect huge amount of visual information. All these raw visual data are often transported over a public network to distant video analytic centres. This increases the risk of interception and the spill of individuals' information into the wider cyberspace that risks privacy breaches. The edge computing paradigm allows the enforcement of privacy protection mechanisms at the point where the video frames are created. Nonetheless, existing cryptographic schemes are computationally unaffordable at the resource-constrained network edge. Based on chaotic methods, three lightweight end-to-end privacy-protection mechanisms are proposed: (1) a novel lightweight Sine-cosine Chaotic Map, which is a robust and efficient solution for enciphering frames at edge cameras; (2) Dynamic Chaotic Image Enciphering scheme that can run in real time at the edge; (3) a lightweight Regions of Interest Masking scheme that ensures the privacy of sensitive attributes like face on video frames. Design rationales are discussed and extensive experimental analyses substantiate the feasibility and security of the proposed schemes.

KEYWORDS

information security and privacy, smart cities applications

1 | INTRODUCTION

The rapid advancement and proliferation of electronic technologies in recent years have driven urban areas to become a lot smarter. Currently, a multitude of cities around the world employ a spectrum of information and communication technologies and Internet of Things to improve the quality of urban services and to ensure the safety and security of their residences [1–3]. Surveillance is often practiced through the use of both fixedly deployed closed circuit television (CCTV) cameras and cameras mounted on mobile manned or unmanned aerial and ground vehicles like aeroplanes, satellites, drones, manned ground patrolling vehicles, and unmanned ground vehicles [4, 5]. The ubiquitous and versatile deployment of these CCTV cameras in public places including streets, city corners, stores, and marketplaces enables the first-responders, government agencies, or security service providers to garner a great deal of audio-visual information about many individuals indiscriminately without their knowledge and consent [6–9].

Up to 2021, there are more than a billion fixed surveillance cameras in operation in urban and suburban areas across the world [10]. All the visual information about many individuals created and collected by these CCTV cameras, which serves as the eyes and ears of the video surveillance system (VSS), are most often transported over a public network to distant video analytics and surveillance operation centres (SOC). This is one of the major factors that increases the likelihood of the breach of individuals' privacy due to some inherent vulnerabilities of the network architecture. The TCP/IP network architecture, widely used in today's Internet, is liable to many threats for it was initially designed without due sufficient concern to security; it is one of the most ingenious human inventions of the 20th century, though [3, 11, 12]. Hence, any deployment of the surveillance system without due attention to such flaws is likely to increase the risk of privacy invasion. Most of today's VSS are deployed based on either the fog or cloud computing architecture. The cloud computing paradigms get the video analytics performed in remote cloud centres that have tremendous computational

capability. However, the raw video streams may be intercepted exploiting the vulnerabilities of the network over which they are channelled. Generally, the fog and cloud computing paradigms increase the probability of privacy breaches.

Meanwhile, the edge computing paradigm, as stated in Ref. [13], migrates some intelligence and computational prowess to the smart cameras allowing end-to-end (E2E) privacy-preserving mechanisms to be enforced at the point where the videos are created. E2E Privacy refers to the scenario where information between two or more communicating parties or end systems is exchanged over secure communication channels that prevent unauthorised parts from accessing the information. In the case of a VSS, the video frames are supposed to be scrambled or encrypted on edge cameras before transmission in order to ensure that only legitimate recipients or devices that hold the unscrambling key are able to reverse the frames. This decreases the chance of divulging of individuals' data into the wider cyberspace where there are more than 4.57 billion users [14]. However, edge computing is a resource-constrained environment and it cannot support most of the well-known encryption methods that are compute-intensive and computer-intensive. Lightweight techniques that balance the privacy preserving requirements and affordable processing complexity are compelling [15].

In this work, we proposed and experimentally validated three lightweight video frame scrambling schemes. They are affordable to edge CCTV cameras to enable E2E privacy in video surveillance systems (VSS). The major contributions of this paper are summarised as ensues:

- A new lightweight Sine-cosine Chaotic Map (SCM) is proposed for faster full video-frame scrambling to ensure E2E privacy in comparison to preexisting schemes. Our experimental results show that the SCM is the most robust and efficient solution for enciphering full frames at edge cameras. Its control parameters have much wider secure operating ranges.
- To further improve the processing speed, a simpler but non-linear lightweight Dynamic Chaotic Image Enciphering (DyCIE) scheme is proposed based on a discrete chaotic dynamic system [16, 17] to ensure E2E privacy, which can run at the edge of the network in real time. A number of control parameters are introduced ensuing a thorough security analysis. The DyCIE scheme is highly sensitive to slight changes in the initial conditions that are used as parts of the key. As a result, DyCIE is more efficient for E2E video encryption in terms of both speed and security.
- Based on the Peter De Jong Map [18], a lightweight Regions of Interest (RoI) Masking (RoI-Mask) scheme is proposed to ensure the privacy of sensitive attributes on video frames. As the original De Jong map is prone to a range of attacks, we introduce additional control parameters through a lot of experimental and fine-tuning works to transform it into a form usable for cryptographic purposes. This is more efficient for scrambling parts of a frame, not full frames.
- In order to verify the computational efficacy and security of the proposed schemes, comprehensive experimental study and analysis have been carried out, including computational performance analysis, standard security analyses, and comparative analyses with existing equivalent methods. The results solidly corroborate the feasibility of our schemes for scrambling full frames and RoIs.

The rest of the paper is organised as follows. Section 2 discusses the related works on video privacy protection schemes and chaotic theories. Section 4 presents a high-level architecture for E2E privacy-preserving scheme in the practice of video surveillance. Section 6 elucidates the bolts and nuts of the computationally-thin DyCIE scheme developed based on a simple logistic map. The SCM scheme for full-frame scrambling is explicated in Section 5 ensued by discussion of the RoI-Mask scheme for masking RoIs on video frames in Section 7. Section 8 presents a frame-shuffling scheme. The detailed experimental analyses and results are presented in Section 9. At last, the conclusions are presented in Section 10.

2 | RELATED WORKS

2.1 | Video privacy-protection techniques

In general, today's video/image privacy protection schemes can be put into four categories, namely *editing, face regions, false colour, and JPEG* [19]. However, apart from the encryption scheme, most of the editing schemes are unable to completely hide sensitive contents on images. Besides, they are prone to reconstruction attacks [20, 21]. They include simple schemes like blurring, black box, pixelation, and masking. But most importantly, information is not fully recoverable. Similarly, the face regions approach [9, 22, 23] suffers from not being appropriate to real-time processing and reversibility. The false colour [24] and JPEG [25, 26] methods also suffer from similar type of problems. They fail to meet the requirements set to achieve a good trade-off between usability and privacy. Ideally, video-contents privacy protection schemes must accomplish a good balance among privacy, clarity, reversibility, and security [27, 28]. Privacy refers to the condition of not being identified by human observers without one's knowledge and consent. Clarity is an important requirement that any privacy protection mechanism should allow the identification of suspicious behavioural patterns. Any privacy-protection scheme should also have a reversibility attribute in order for privacy-protected frames that contain crime scenes or criminal activities to be reversible. In addition, the scheme must be secure and robust. The security property ensures that the encrypted video frames are reversed only by authorised parties.

Encryption is a member of the *editing* class video-privacy protection schemes. Given the aforementioned requirements, encryption/scrambling is the most secure video privacy protection scheme compared to the other editing schemes. It protects private information on video frames from

unauthorised accesses by scrambling the sensitive features before transmission. This way, it ensures E2E privacy of the communication between two communicating parties. There are plenty of encryption schemes today; however, most of them are not suitable for encrypting real-time video frames. Given the real-time nature and bulkiness of videos, public-key cryptographic schemes like Rivest, Shamir and Adleman (RSA) and Elliptical Curve Cryptography (ECC) are too slow to be employed in such scenarios. The traditional symmetric key cryptographic mechanisms like triple Data Encryption Standard (3DES), International Data Encryption Algorithm (IDEA), the malleable and fast Rivest Cipher 4 (RC4) and Advanced Data Encryption chaining block cipher (AES-CBC) are not convenient for image enciphering, either. AES is considered as one of the most secure ciphers commonly used in the secure sockets layer (SSL) or transport layer security (TLS) across the Internet today. However, it is relatively slow to be employed for real-time video encryption at the edge, where there is a limited computational power. RC4 is a quick stream cipher but it is not considered as a secure cipher any longer because of its vulnerability to a bit-flipping attack where one in every 256 keys can be a weak key. In addition, as shown by the previous studies [29, 30] as well as by our own evaluation in Section 9.5, both AES and RC4 are not good in breaking the strong correlation amongst horizontally, vertically, and diagonally adjacent pixels of video frames.

The most popular schemes for image encryption are those mechanisms created based on chaotic schemes because of their better performance and high security. Unlike AES, the chaotic schemes can break the strong correlation amongst the pixel values of information-rich images/frames. There are chaotic maps of different dimensions including one dimensional, multidimensional and cascaded chaotic systems. There are many good chaotic systems proposed by a number of researchers [17, 31–34]. More recently proposed chaotic image enciphering mechanisms presented in Refs. [35, 36] are robust. However, they are still compute-intensive and cannot achieve the required performance in resource-constrained environments, like the edge of a network. They must be customised to fit into edge devices; or new lighter methods must be introduced. Hence, this work aims at proposing novel lightweight video-frame scrambling schemes for secure and efficient enciphering of video frames at the edge of networks to ensure E2E privacy vis-à-vis the practice of VSS.

2.2 | Chaotic theory

Literally, chaos is a state of complete disorder and confusion. In chaos theory, a chaotic system is a dynamical system that is highly sensitive to initial conditions, and one which is topologically mixing. Chaos can be produced either by using uncontrolled or controlled systems, as portrayed in Figure 1. The ones in Figure 1a are produced by uncontrolled natural processes. The top one is a picture of the turbulence in the tip vortex from an aeroplane wing and the bottom one is a random fractal of lightning scene. Figure 1b illustrates a 10-scroll chaotic-attractor produced by a controlled chaotic generator that we worked on previously [31, 38].

Chaotic techniques are the most useful ones in video frames/images scrambling because they have better performance, high degree of sensitivity to slight changes in initial

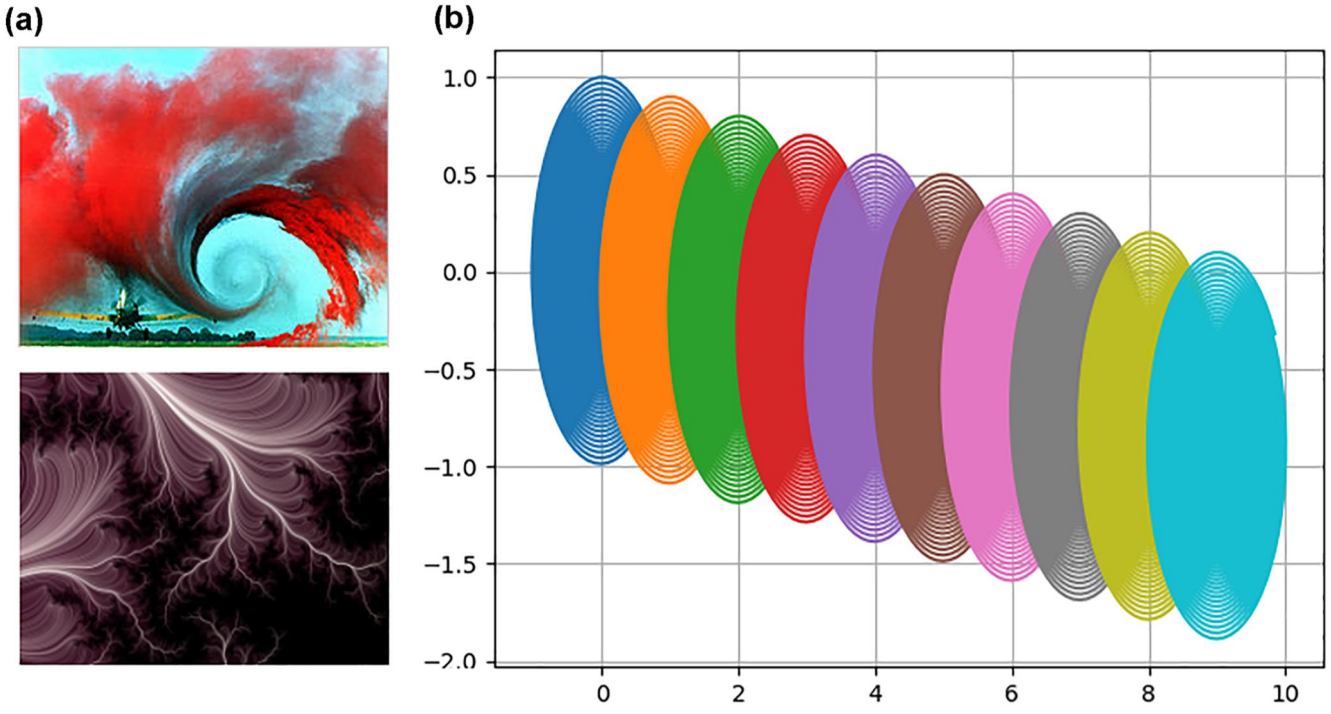

FIGURE 1 Chaos: (a) chaos produced by uncontrolled natural processes [37], (b) chaos produced by a controlled system

conditions, higher degree of randomness, enormous key space, and high security. Particularly, they are able to dissociate the strong correlation among the adjacent pixels of a frame. Besides, a video frame/image contains bulky information; as a result, the traditional methods are slow to be used to encrypt such bulky data in a time sensitive application. Another most outstanding advantage of employing chaos for scrambling video frames in lieu of other methods is the fact that it can be easily *vectorised*. The vectorisation process is employed to speed up the implementation of bulky data represented in the form of 2D or 3D matrices or rank 2 or 3 tensors, as portrayed in Figure 2, without using a computationally expensive looping structure. Once the chaos is produced, its 3D matrix of pixel values, shown in Figure 2, can be matched with the corresponding pixels of the frame and operated in parallel. Using such an approach can drastically improve the computational performance.

Equation (1) illustrates the vectorised bit-wise-xor operation between two RGB coloured images (a chaos and video frame), whose height, width, and depth are given to be H, W, and M, respectively. Putting it another way, the pixels in corresponding locations are operated in parallel unlike the serial bit-wise-xor operation where only a pair of corresponding pixels are operated at a time.

$$I_1 \oplus I_2 = \sum_{i=0}^{W-1}\sum_{j=0}^{H-1}\sum_{k=0}^{M-1} I_1(i,j,k) \oplus \sum_{i=0}^{W-1}\sum_{j=0}^{H-1}\sum_{k=0}^{M-1} I_2(i,j,k) \tag{1}$$

Table 1 compares the difference in performance between serial and vectorised bit-wise-xor operations. Two 3-channel coloured 480P (640 × 480 × 3) frames/images were employed for the experiment. The results in Table 1 shows that the vectorised operator is about 153 times faster than its serial version. This, along the good security properties, lays a strong rationale for employing chaotic schemes for image scrambling to ensure E2E privacy. However, most preexisting chaotic schemes suffer from higher computational complexity. They are not suitable for deployment in edge cameras which have very limited computational resources. This work, therefore, focuses on designing, testing, and implementing lightweight and secure chaotic methods for video frame scrambling at the edge of the network.

3 | AN ATTACKER MODEL

Vis-à-vis the schemes proposed in subsequent sections and their security analysis, the attacker model considered is portrayed in Figure 3. Normally, the intruder or attacker can only access the enciphered frames easily while in transit from the point of creation (edge-camera) to analytics centre or storage site at fog/cloud server, and then to the surveillance operation centres (SOC) where the authorised security personnel or law enforcers sit to remotely observe the activities of individuals caught on CCTV cameras. Hence, the attacker is not assumed be able to obtain any information from the cipher that can identify individuals caught on the original frame. If frames are transmitted in a raw form or the encryption mechanism is weaker, the intruder can access the contents of frames and spill them in the wider cyberspace where there are more than 4.6 billion active users at present.

However, for the sake of testing the security of the proposed mechanisms, the attackers in this testing model are assumed to have an access to a plain frame and its corresponding cipher so that they can perform visual assessment, differential analysis, histogram/frequency analysis, entropy analysis, and correlation analysis on the cipher and its corresponding clear form to find clues or weaknesses in the proposed algorithms.

$[11]$	$\begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_m \end{bmatrix}$	$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$	$\begin{bmatrix} \begin{bmatrix} a_{111} & a_{121} & \dots & a_{1n1} \\ a_{211} & a_{221} & \dots & a_{2n1} \\ \dots & \dots & \dots & \dots \\ a_{m11} & a_{m21} & \dots & a_{mn1} \end{bmatrix} & \begin{bmatrix} a_{112} & a_{122} & \dots & a_{1n2} \\ a_{2n2} \\ \dots \\ a_{mn2} \end{bmatrix} & \begin{bmatrix} a_{113} & a_{123} & \dots & a_{1n3} \\ a_{2n3} \\ \dots \\ a_{mn3} \end{bmatrix} \end{bmatrix}$
Scalar	*Vector*	*Matrix*	*Tensor*
$Rank_0$	$Rank_1$	$Rank_2$	$Rank_3$
Tensor 1×1	*Tensor* $m \times 1$ *or* $1 \times n$	*Tensor* $m \times n$ *2D Image*	*Tensor* $m \times n \times k$ *3D Image*

FIGURE 2 Tensors of various ranks

Parameter	Serial bitwise-xor	Vectorised-xor
Time (ms)	456.863	2.992
Relative performance	0.00654 times as fast as the vectorised xor	152.7 times faster than the serial xor

TABLE 1 Comparing serial and vectorised bit-wise-xor operator

4 | PRIVACY-PRESERVING SMART EDGE SURVEILLANCE

Figure 4 shows a high level overview of the architecture of the smart VSS at the edge, in which our proposed video-frame scrambling mechanisms are envisioned for ensuring E2E privacy. There are five major components, namely edge camera, fog/cloud server, storage site, SOC, and law enforcer, involved in the VSS architecture. Hence, the E2E privacy ensures that no one can eavesdrop on the contents of video frames while they are in transit. No adversary can snoop on the video frames without authorised permissions. This helps prevent the breaches of individuals' privacy by wiretapping or interception attacks. Cryptographic methods are, therefore, employed to ensure E2E privacy. In other words, video frame encryption is of paramount importance in ensuring E2E privacy in VSS. More details on the design rationales, function blocks, and evaluation reports of the edge VSS is beyond the scope of this study, interested readers can find details in literature [15, 38–44]. This study is about mechanisms designed to ensure E2E privacy in VSS, not about the surveillance per se.

Privacy-protection schemes must be secure in that the reversion is done only by authorised parties who possess the right set of keys. A robust but lightweight key distribution management scheme is vital specifically in our E2E privacy-preserving schemes. Therefore, a simplified key management has been introduced. As portrayed in Figure 4, video streams are transmitted from cameras to the fog/cloud server, from the fog/cloud server to the viewing stations, or from the fog/cloud server to storage locations in encrypted form. Hence, a client-server architecture-based lightweight agents are employed for efficient key distribution. The key is generated in the form of list data structure, $Key = [K_R, K_G, K_B]$, where K_R, K_G and K_B are the set of keys employed for the enciphering of the three channels of the input frame. Camera, server, storage site, and viewing-station agents are developed and deployed.

The camera agent stores the public key of the server agent, and the server agent keeps record of the public keys of the viewing-station and storage agents for the purpose of secure session key exchange. The viewing-station agent unscrambles frames to be viewed live by security personnel in a SOC.

5 | SINE-COSINE CHAOTIC MAP (SCM)

A new lightweight SCM is proposed for video frame encryption on edge devices based on a chaotic, statistical and security analysis. The output of the SCM is determined by the values of its control parameters/coefficients (α, β and γ) and an initial condition (x_0) as clearly defined by Equation (2). These parameters play very pronounced roles in making the cipher diffused and confusing enough and they constitute the encryption-decryption key of this scheme.

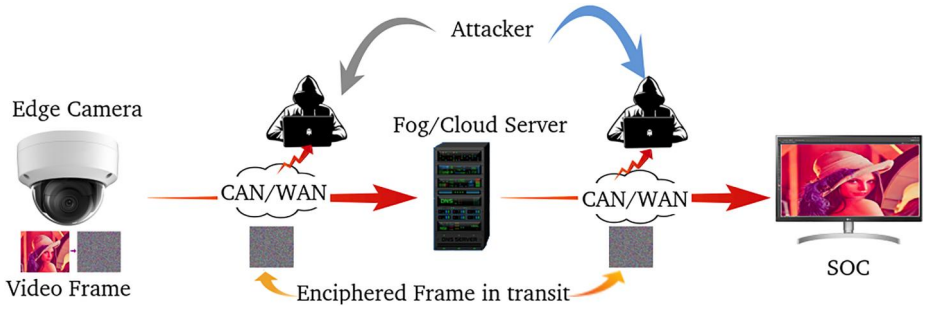

FIGURE 3 Attacker model comprising an edge-camera, an intruder, a fog/cloud server, and surveillance operation centre

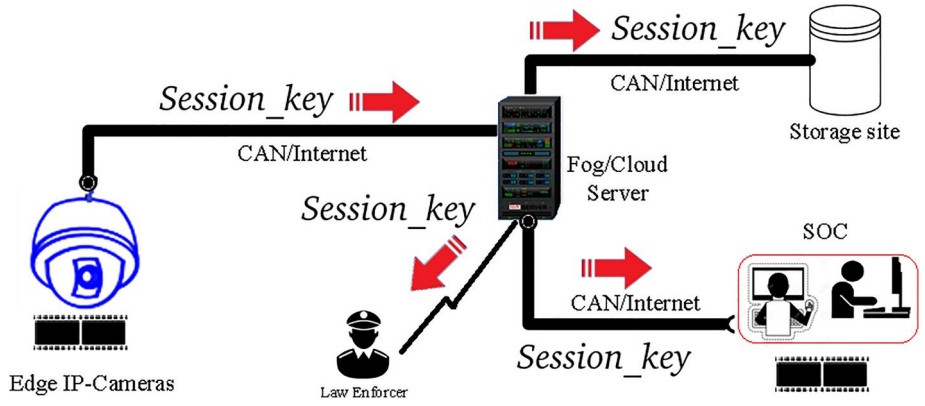

FIGURE 4 Simplified end-to-end privacy diagram of agents interaction for key exchange

$$x_{n+1} = \alpha \sin(\beta x_n) + 0.5 \times \alpha\gamma(1 - \cos(2\beta x_n)) \quad (2)$$

Following a thorough analysis of Equation (2), a multiplying constant was added to the system to further improve the security of the chaotic sequence generator. It generates a secure and evenly distributed chaotic image. Algorithm 1 basically describes how the key and chaos are generated with three more parameters added to those stated in Equation (2). They are a multiplier of value 1624, a scaling parameter θ and an updating parameter δ.

Algorithm 1 Major steps

```
chaos ← [][]
Key Generation:
Key := [x0, α, β, γ, δ, θ]
Sequence Initialisation:
xn = x0
Generate Chaos of size of size W × H
tmp1 = α × sin(β × xn)
tmp2 = 0.5 × αγcos(2β × xn)
const = 0.5αγ
xn+1 = 1624[tmp1 − tmp2 + const]%1
chaos ← xn+1
Update xn:
xn = δ × xn+1
```

The video frame scrambling using the sinusoidal chaotic sequence is performed colour-channel-wise simultaneously to reduce the processing time. The image scrambling process comprises a key-generator and chaos-generator modules, as described in Algorithm 1. The key generator module (key_gen ()) generates a key comprising six elements indexed from $i = 0$ to $i = 5$, each 64-bit long, as stated in Equation (3).

$$\begin{aligned} Key &= gen\_key() \\ &= [x_0,\ \alpha,\ \beta,\ \gamma,\ \delta,\ \theta] \end{aligned} \quad (3)$$

where x_0 is the initial condition of the system that triggers the chaotic generator to recursively generate the required set of chaotic pixels. Then, the respective keys for the enciphering of the three colour channels of the image are generated by using Equation (4).

$$\begin{aligned} Key_r &= [x_0,\ \alpha_r,\ \beta_r,\ \gamma_r,\ \delta_r,\ \theta_r] = gen\_key() \\ Key_g &= [x_0,\ \alpha_g,\ \beta_g,\ \gamma_g,\ \delta_g,\ \theta_g] = gen\_key() \\ Key_b &= [x_0,\ \alpha_b,\ \beta_b,\ \gamma_b,\ \delta_b,\ \theta_b] = gen\_key() \end{aligned} \quad (4)$$

$r_0, g_0, b_0 = Key_r[0], Key_g[0], Key_b[0]$ where r_0, g_0, and b_0 are the initial values for channels R, G, and B, respectively. Every key element is a 64-bit floating point value randomly picked from a domain in the range (a, b), where there are infinitely many real-valued numbers between a and b. To enable parallel encryption of the colour channels of every frame, the required chaotic images ($Chaos_r$, $Chaos_g$, *and* $Chaos_b$) are produced by using Key_r, Key_g, *and* Key_b

In a more elaborate manner, Algorithm 2 describes the chaos generation process by using the SCM method, where the channel keys, chaos, and scrambling processes are performed in parallel. The chaos size is determined by the size of the frame ($W \times H$) to be encrypted. That is, the frame is first shuffled by the algorithm described in Section 8 and then xored with the chaos generated by using the steps stated in Algorithm 2.

Algorithm 2 Frame scrambling using SCM

```
1: procedure GEN_KEY
2:     key ← Equation 3
3:     return key
4: procedure GEN_CHAOS(key)
5:     chaos ← Equations 2, 4
6:     return chaos
7: procedure SCRAMBLE_FRAME(frame, chaos)
8:     frame ← shuffle(frame)
9:     frame_enc ← frame vectorised − ⊕chaos
10:    return frame_enc
11: procedure UNSCRAMBLE_FRAME(frame_enc, chaos)
12:    frame_clear ← frame_enc ⊕ chaos
13:    return frame_clear
14: At Sending End (Edge Camera)
15: vid ← videoCapture()
16: W,H ← 640,480
17: while True do
18:    status,frame ← vid.read()
19:    frame ← frame.resize((W, H))
20:    if status then
21:        key ← gen_key()
22:        chaos ← gen_chaos(key)
23:        f_enc ← scramble_frame(frame, chaos)
24: At Receiving End (Server, SOC,or Storage
    sites)
25: while True do
26:    f_enc, key ← from sender (camera)
27:    chaos ← gen_chaos(key)
28:    f_clear ← unscramble_frame(f_enc, chaos)
29:    f_clear ← unshuffle(f_clear)
```

6 | DYNAMIC CHAOTIC IMAGE ENCIPHERING SCHEME

In this section, a lightweight DyCIE scheme is proposed as an Improved Logistic Map (ILM). Developed after thoroughly investigating the simpler discrete chaotic dynamic systems [16, 17], DyCIE scheme can run at the edge in real time. It is lighter and highly sensitive to any slight variation in the values of a key and more efficient for video encryption. The encryption key is defined as a list data type in Equation (5) and its elements constitute the coefficients and initial value of the chaotic system.

$$\begin{aligned} K &= [k_0,\ k_1,\ k_2,\ k_3,\ k_4] \\ C_0 &= k_0 \\ tmp &= k_1 \times C_0(1 + k_2 \times C_0) \\ C &= tmp \times k_3\ (scaling) \\ C_0 &= tmp \times k_4\ (update) \end{aligned} \tag{5}$$

where $k_0 \in (0.2, 0.9)$, $k_1 \in (3.89, 4.0)$, $k_2 \in (-1, -0.989)$, $k_3 \in (254, 255)$, and $k_4 \in (0.99, 1)$.

Equation (5) is recursive in that the new chaotic values of the dynamic system in the equation are generated based on previous values multiplied by some control parameters. It starts with an initial value C_0 and coefficients k_1 and k_2. Then, it is iterated until a maximum number of iterations equal to the product of the height and width of an input frame is reached. Extensive experiments and analyses have been carried out to obtain the secure ranges of the initial value k_0, and the coefficients k_1, k_2, k_3, and k_4, which are generated in the form of encryption key. Based on our experimental study, k_0 can take any double-precision floating value between (0.2, 0.9). For k_1, the secure range that generates secure random chaos is found out to be between (3.89, 4.0). k_2 and k_4 can assume floating values in the range of (0.99, 1.0) and k_3 must be within the range of (254, 255).

As outlined by Algorithm 3, the DyCIE scheme comprises three major components: key generator, chaos generator, and frame scrambler. The key is defined as a list data type whose elements are the initial value and the coefficients of the enciphering equations. Their values fall within the defined secure ranges and are generated using a secure cryptographic random generator. The chaos generator, which takes the key and video-frame dimensions as inputs, produces a random chaos that in turn is used as a key to encipher frames.

Algorithm 3 DyCIE

```
procedure KEYGEN
    K ← secRand([k0, k1, k2, k3, k4])
    return K
procedure GENCHAOS(K, w, h)
    i ← 0
    C0 ← K[0]
    while True do
        temp ← K[1] × textC0(1− K[2] × C0)
        C[i] ← ceil(temp ×K[3])
        C0 ← temp × K[4]
        i+ = 1
        if i = = (h ×w) then
            break
    return C
procedure ENCFRAME(C, fc)
    fenc ← fc ⊕ C
    return fenc
procedure DECFRAME(C, fenc)
    fc ← fenc ⊕ C
    returnfc
At Sending End
fc ← cam.video()
C ← []
w,h ← fc.size
K ← keyGen()
C ← genChaos(K, w, h)
fenc ← encFrame(fc, C)
At Receiving End
K, fenc ← received from sending end
w,h ← fenc.size
C ← genChaos(K, w, h)
fc ← decFrame(fenc, C)
```

To enhance the security of DyCIE scheme and to cut down on its computational complexity, the enciphering process is performed channel-wise. As portrayed in Figure 5, the frame to be encrypted is efficiently split into the three colour channels, namely red (R), green (G), and blue (B). Then, the three colour channels are encrypted in parallel and the results are stacked up together before transmission. The three keys are also appended

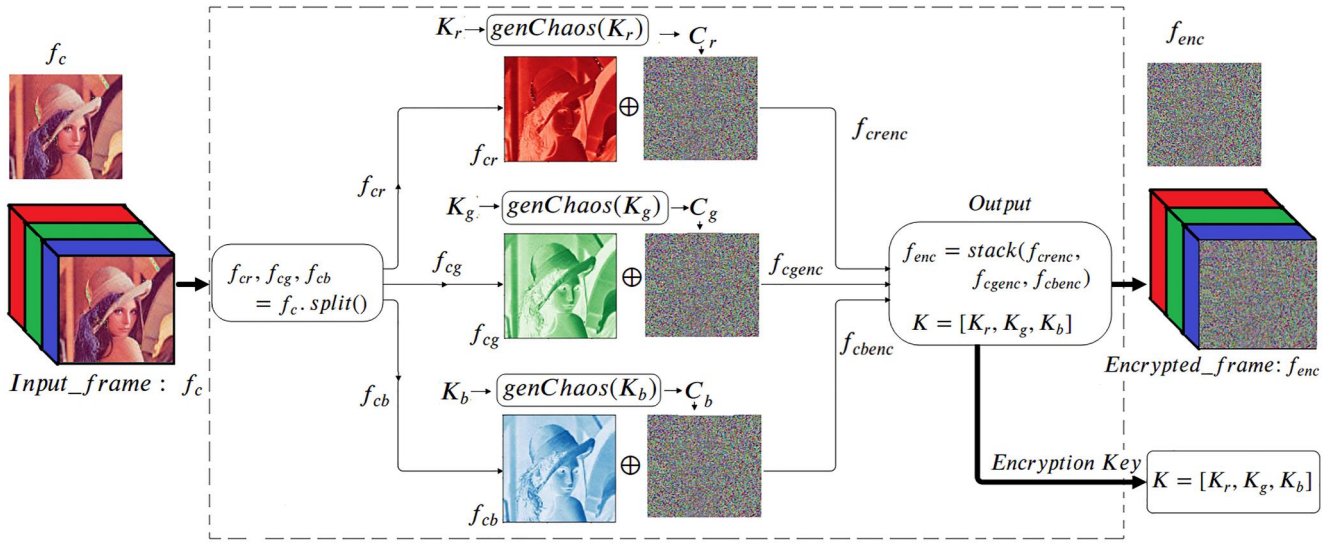

FIGURE 5 Channel-wise Frame Enciphering: colour channels R, G, and B are scrambled in parallel in order to expedite the scrambling process

into a single list before transmission in the same order as the colour channels (RGB). Figure 5 illustrates that the enciphering process is performed at the edge and the inverse process is performed at the receiving end as shown in Section 4.

7 | REGIONS OF INTEREST MASKING

Unlike the SCM and DyCIE methods, the lightweight Regions of Interest Masking (RoI-Mask) scheme was developed based on the Peter De Jong Map [18] specifically for protecting privacy-sensitive objects on frames. The Peter De Jong map is a type of 2D recursive system stated in Equation (6). The choice of parameter values and initial conditions will generate totally different sets of attractors. It has four parameters and two initial conditions. However, the Peter de Jong map cannot be used for chaotic encryption as is. It does not meet the security requirements. For instance, Figure 6 shows a non-uniform distribution of the pixels of chaos generated by Equation (6), signifying that it is glaringly insecure.

$$\begin{aligned} x_{n+1} &= \sin(a \times y_n) - \cos(b \times x_n) \\ y_{n+1} &= \sin(c \times x_n) - \cos(d \times y_n) \end{aligned} \tag{6}$$

Following an extensive experimental study, RoI-Mask is proposed as a one-way scrambling of RoI in video frames using an improved version of De Jong map (IDJM) and vectorised pixel-array multiplication. As illustrated in Equation (7), four more parameters are added and their secure ranges are identified. It generates a random and uniform output that is secure to be used for cryptographic purpose. A key comprising eight elements is first generated followed by a random chaos used to scramble the RoIs on video frames. The top five lines of Equation (7) illustrate the secure range of the eight elements of the key employed to generate chaos equal to the size of the RoI. Every element of the key is a double-precision floating-point value randomly selected from its respective range, which were identified after an extensive experimental and security analysis. These values allow to produce secure chaotic outcomes that are of the same size as the RoI, and a vectorised point-wise multiplication operation is performed, which results in secure cipher.

$$\begin{aligned} key &= [k_0,\ k_1,\ k_2,\ k_3,\ k_4,\ k_5,\ k_6,\ k_7 \\ k_0,\ k_1 &= random(1,4),\ random(1,4) \\ k_2,\ k_5 &= random(-1,1),\ random(-1,1) \\ k_3,\ k_6 &= random(-1,-0.99),\ random(-1,-0.99) \\ k_4,\ k_7 &= random(-3,-3),\ random(-3,-3) \\ x_0,\ y_0 &= k_0,\ k_1 \\ x_1 &= \sin(k_2 * y_0) + k_3 * \cos(k_4 * x_0) \\ y_1 &= \sin(k_5 * x_0) + k_6 * \cos(k_7 * y_0) \\ x_0,\ y_0 &= x_1,\ y_1 \end{aligned} \tag{7}$$

Algorithm 4 Region-of-Interest Masking Algorithm

```
RoI ← Object - dtector(cam.video())
procedure GENERATEKEY
    K ← secRand([k0, k1, …, k9])
    return ← K
key ← generateKey()
procedure GENERATEKEY(key, W, H)
    chaos ← Equation 7
    return ← chaos
chaos ← generateKey(key)
procedure DENATUREFACES(chaos, RoIxy)
    f_denatured ← vectorised_multiply(RoI, chaos)
    return ← f_denatured
```

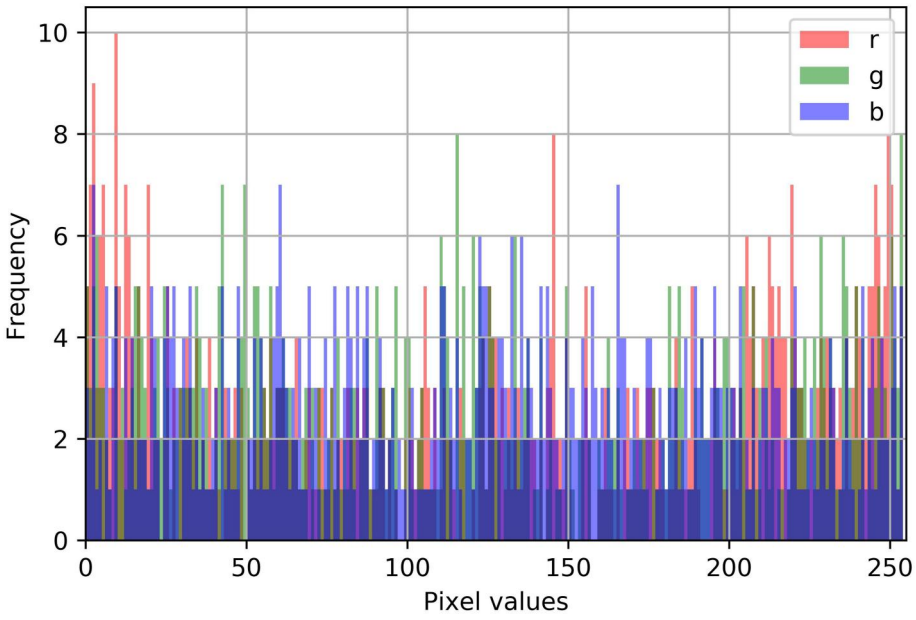

FIGURE 6 Distribution of the De Jong Chaos: it is not uniform revealing its weakness against histogram analysis attack

Algorithm 4 shows that a vectorised multiplication is employed. This algorithm is suitable for anonymising only small regions on a frame. In our work, we employed it for denaturing a specific RoI, like faces, which usually is a small portion of a whole frame. However, such an approach is not quite ideal for full frame scrambling. In scenarios where there is a need to ensure E2E privacy for the entire frame, this is not the way to go because multiplication normally requires more time (it takes more gates) than other simpler but more efficient operators like bitwise XOR.

8 | FRAME SHUFFLING TO ENHANCE SECURITY

In this section, we have introduced a simple but efficient frame shuffling algorithm so as to further improve the security of the proposed frame encryption schemes. It shuffles the pixels of a frame in a block size of $n \times n$. This increases the diffusion of pixels or block of pixels which increases clear-frame sensitivity important for cutting down on the probability of differential attacks. For a better trade-off of speed and security, a block size of 32×32 is employed in this work. The procedure is depicted in Algorithm 5 pythonically.

As pythonically depicted in Algorithm 5, all possible block (x, y) positions are combined together as tuples in a list using list comprehension and the *zip* method. Then, they are truly shuffled using a data frame after an index has been added. The shuffling method employed (Fisher–Yates) is inherently irreversible; however, we employed the concept of index to reproduce the original tuple positions. The block size, and *index_key* employed to shuffle the input frame at the sending end must be securely forwarded to the receiving end along with the key. At the receiving end, the original positions of the blocks in the frame are restored by simply sorting the *index_key* column. Figure 7a shows the original positions of blocks in an input frame, and (b) portrays the randomised positions $((x, y) \leftarrow (h, w))$ of the blocks along with their respective *index_key* values at the sending end. At the receiving end, Figure 7c shows the restored positions of blocks in the received frame. Algorithm 5, is a comprehensive python pseudocode that lists methods, data structures, numpy methods and attributes (shape, reshape, and *r_*), and iteration tools.

Algorithm 5 Frame pixels shuffling (pythonic)

```
1: import random
2: from numpy import r_
3: import itertools as it
4: import numpy as np
5: import pandas as pd
6: W, H ←640, 480
7: procedure SHUFFLE_FRAME(frame, blk_size)
8:   h ← [i for i in range(0, H, blk_size)]
9:   w ← [j for j in range(0, W, blk_size)]
10:   hw ← list(it.product(h, w))
11:   dfhw ← pd.DataFrame
(hw, columns = ['h','w'])
12:   dfhw['i'] ← lst
13:   dfhw ← dfhw.sample(frac = 1)
14:   hws ← list(zip(dfhw['h'].tolist(),
dfhw['w'].tolist()))
```

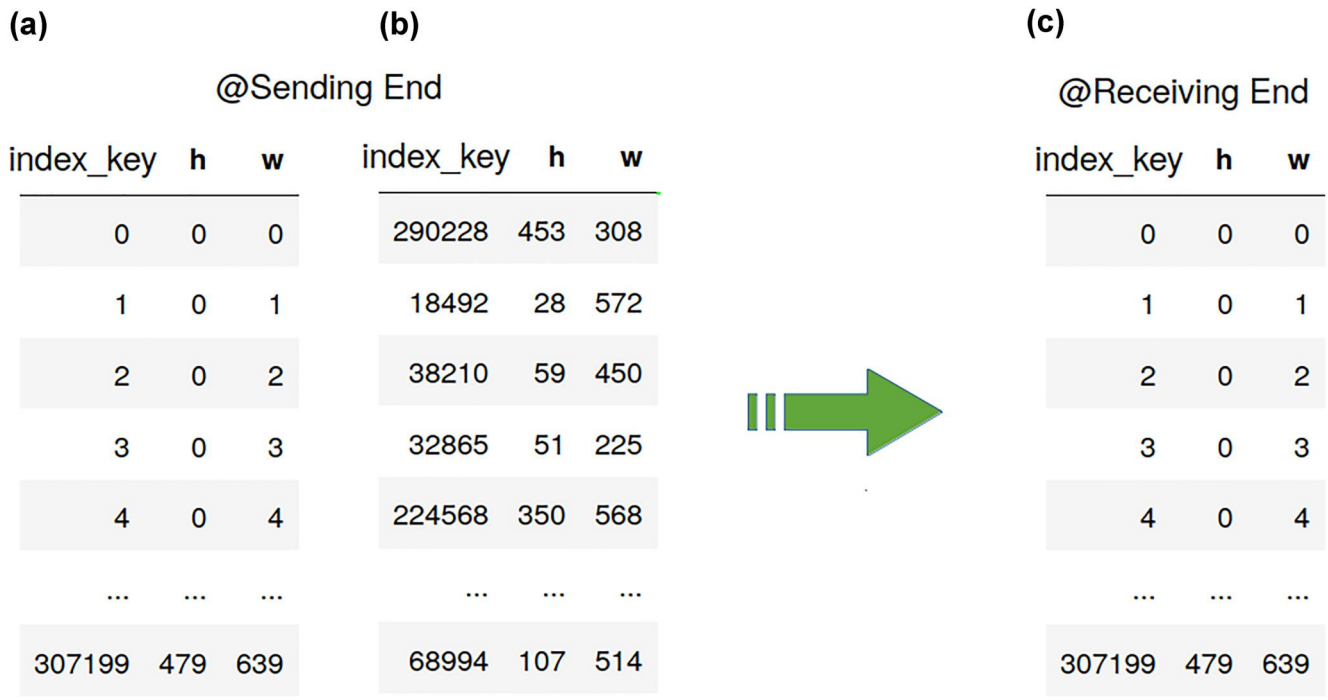

FIGURE 7 Frame Shuffling and Unshuffling: (a) Original positions of blocks and (b) shuffled positions of blocks at sending end, and (c) restored positions of blocks at receiving end

```
15:   index_key ← dfhw.index.tolist()
16:   type ← np.asarray(hws,
dtype = [('x', 'i4'), ('y', 'i4')]
17:   hws ← type, order = 'C')
18:   hws ← hws.reshape(int(H/
blk_size), int(W/blk_size))
19:   imsize ← ch.shape
20:   imgn ← ch
21:   c1 ← 0
22:   for i in r_[: imsize[0]: blk_size] do
23:     c2 = 0
24:     for j in r_[: imsize[0]:
blk_size] do
25:       x, y = hws[c1][c2]
26:       imgn[i: (i + blk_size), j:
(j + blk_size)] =
27:       ch[x: (x + blk_size), y:
(y + blk_size)]
28: c1+ = 1
29:   return ch, index_key
```

9 | EXPERIMENTS AND RESULT ANALYSIS

In the experimental setup, smart CCTV cameras deployed at the edge of the network with a Raspberry Pi 4 incorporated into them, whose specifications are provided in Table 2, are employed. These low-cost, tiny single board computers (SBC) are vital for enforcing the privacy-preserving mechanisms on the video frames at the point of creation to ensure E2E privacy protection. The implementation is done using Python 3.7.4 with multithreading and multiprocessing enabled. Besides, the video-frame employed for the experimental analysis and implementation has a size of 480P (480 × 640 × 3) and three colour channels (RGB) with a pixel depth of 8 bits.

In addition to performance analysis, a comprehensive experimental study has been conducted considering the attack model in Section 3, including a comparative study of our proposed schemes (SCM, DyCIE, and RoI-Mask) with two classic solutions (RC4 and AES) plus three recently reported efforts. Several cases and computational-and-statistical security parameters were employed to analyse the performance and security of the proposed schemes. The properties, computational and statistical security parameters considered include Lyapunov Exponents, visual assessment, key space and chaos size analysis, key sensitivity analysis, statistical analysis, Peak Signal to Noise Ratio (PSNR), Number of Pixels Change Rate (NPCR), Unified Average Changing Intensity (UACI), histogram analysis, correlation analysis, differential analysis, and Information Entropy analysis. All these tests are demonstrated in detail for one of the schemes and the results of all methods analysed and compared are summarised in Table 4, in Subsection 9.5.

TABLE 2 Specification of the Raspberry Pi 4

Parameters	Specifications
CPU type/speed	Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5 GHz
RAM size	4 GB LPDDR4-2400 SDRAM
Integrated Wi-Fi	2.4 and 5 GHz
Ethernet speed	1 Gbps
Camera port	2-lane MIPI CSI
Bluetooth	5.0
Power requirement	3 A, 5 V
Operating system	Debian Linux 10-based

9.1 | Sine-cosine Chaotic Maps (SCM) scheme

A thorough experimental study was conducted on the novel SCM scheme with detailed analyses based on standard performance, computational and statistical security parameters. Then, the results have clearly validated its security and computational efficacy.

9.1.1 | Performance analysis

The functionality and the time complexity of the proposed scheme (SCM) are checked, computed and measured in this subsection.

A. Functional test: As portrayed in Figure 8, the input frame (a) is successfully scrambled into a totally random cipher using the SCM method proving its functionality. The inverse process of scrambling also works correctly.

B. Time complexity: Video frames are bulky, and a good-performing scrambling technique is expected to take less time to encrypt a frame. The computational speed is not only affected by the algorithm design but also by other multiple factors, including the type of hardware platform and the programming language employed. In this work, Raspberry PI 4 is used as the edge device which supports *array and vector processing*. Unlike a scalar processor which operates on a single scalar value at a time, array/vector processors can operate on a vector/1D array of values at a time. Hence, in our design, an entire column of a tensor representation of a video frame is processed at a time.

The keys and chaos sets are generated in parallel, and a whole column vector is operated at a time. Besides, the scrambling processes are performed channel-wise in parallel using array/vector processing. Hence, the computational complexity of our proposed scheme, SCM, is provided in Equation (8). Let us say N is the number of all pixels in a 3-channel RGB-frame, R and C are the number of rows and columns, respectively, in the 3D-tensor representation of the video frame. The time complexity of our scheme is linearly

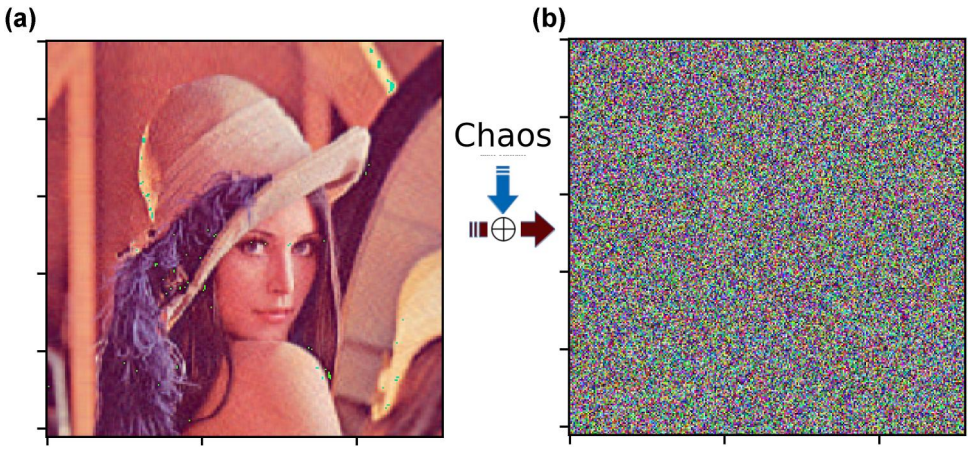

FIGURE 8 Frame scrambling: (a) a clear image; (b) its cipher

proportional to the number of rows (R) in a frame, as depicted in Equation (8).

$$\begin{aligned} Time\_complexity &= \frac{N}{3} \times \frac{1}{C} \\ &= \frac{3 \times R \times C}{3} \times \frac{1}{C} \\ &= R \\ O(N) &= R \end{aligned} \tag{8}$$

C. Encryption time: The corresponding pixels of the clear frame and chaos are mixed in parallel using vectorised *xor*. This approach speeds up the scrambling processes. Our SCM scheme, therefore, can scramble 10.05 frames per second (fps); that is, it takes 95.52 milliseconds to encrypt a single frame.

9.1.2 | Security analysis

Ten computational and statistical security parameters are employed in this subsection to prove the security of the proposed scrambling schemes against visual, statistical, computational, and differential attacks.

A. Lyapunov exponents analysis: The Lyapunov exponents (LE) measure the predictability and sensitivity of a system to changes in its initial conditions, often termed as stability. They are the average logarithmic rate of separation or convergence of two nearby points of two time series X_n and X_{n+1}, separated by an initial distance computed by using Equation (9). LE indicates how evenly the points of a chaotic sequence generator are distributed. It is computed by using Equation (10).

$$\Delta X = \|X_{n+1} - X_n\|^2 \tag{9}$$

$$LE = \lim_{N \to +\infty} \frac{1}{N} \sum_{n=1}^{N} \log\left|\frac{\partial x_{n+1}}{\partial x_n}\right| \tag{10}$$

The Lyapunov exponents of SCM method is computed as follows based on Equation (10).

$$\frac{dx_{n+1}}{dx_n} = \frac{d[\alpha \times \sin(\beta x_n) - tmp \times \cos(2\beta x_n) + tmp]}{dx_n} \tag{11}$$

where $tmp = 0.5 \times \alpha \times \gamma$. Then, the derivatives on Equation (11) were performed by using the calculus product and chain rules that resulted in Equation (12).

$$\frac{dx_{n+1}}{dx_n} = \alpha\beta\cos(\beta x_n) + \alpha\beta\gamma\sin(2\beta x_n) \tag{12}$$

By substituting $\frac{dx_{n+1}}{dx_n}$ in Equation (10) with its expression given by Equation (12), the final formula for computing the Lyapunov exponents of our SCM scheme is obtained as Equation (13). Figure 9 presents the Lyapunov Diagram of the proposed Sinusoidal Map generated by using Equation (13).

$$LE = \lim_{N \to +\infty} \frac{1}{N} \sum_{n=1}^{N} \log|\alpha\beta[tmp_1 + tmp_2]| \tag{13}$$

where $tmp_1 = \cos(\beta \times x_n)$ and $tmp_2 = \gamma \times \sin(2\beta \times x_n)$.

Figure 9 demonstrates that Lyapunov exponents of the sine-cosine map are evenly distributed over unrestricted range. For illustration purpose, the control variable of SCM model is fixed at 20 but it can be extended to any greater value. The Lyapunov exponents vary within 4.36 and 4.45 signifying the uniformity of the proposed SCM scheme. Meanwhile, the popular Logistic map [17] has uneven Lyapunov properties and experiences chaotic properties only within a restricted range of the control variable, between 3.57 and 4. Besides, the Lyapunov exponents of the logistic map vary between −4 and 1 as depicted in Figure 10 indicating non-uniformity. Comparing Figures 9 and 10 clearly shows that SCM scheme achieves a much more even distribution.

B. Visual assessment: A scrambling scheme is said to be good if there is no recognisable visual information on the

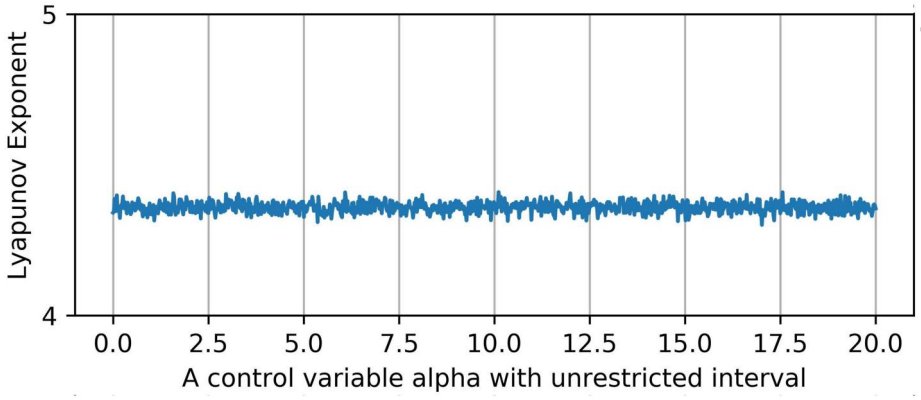

FIGURE 9 Lyapunov diagram of the proposed sinusoidal map

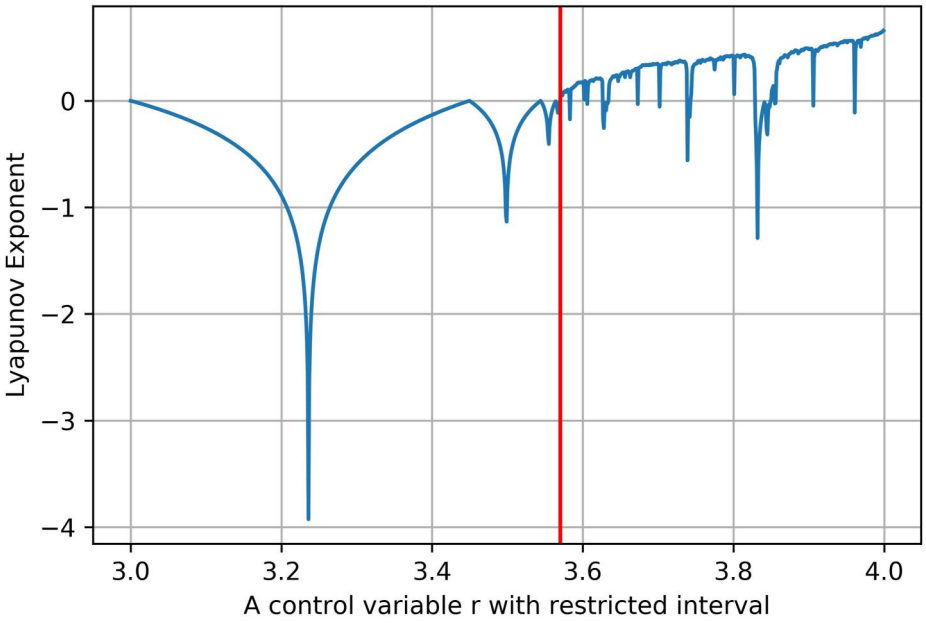

FIGURE 10 Lyapunov diagram of the logistic map [17]

cipher image. As illustrated in Figure 8, the scrambled frame, (b), contains no visually recognisable information about the clear frame, (a). The encrypted image is random and highly disordered proving that the SCM scheme is resistant against any visual assessment attack. This test was conducted on multiple frames.

C. Key space analysis: In the common practice of cryptography, the security of a scheme depends on the length of the keys. It should be sufficiently large and unbreakable by any brute force analysis in a reasonably short period of time. Our SCM scheme has a key that comprises six parameters, each with a 64-bit long double floating-point decimal value. Hence, the total key space of SCM scheme is 2^{384}, which is sufficiently large to resist any possible exhaustive key search analysis attack. Actually, it is much longer than the lower boundary of a secure key space in the practice of symmetrical cryptography, 2^{128}.

D. Histogram analysis: Histogram is the frequency description of each pixel value of a frame. The histograms of the scrambled frames must be totally statistically different from that of corresponding original images. Besides, the scheme is said to be resistant against statistical histogram attacks if it has uniform histograms.

Figure 11 shows the histograms of the unscrambled frame channels and their corresponding scrambled versions. The histograms of the scrambled versions shown in Figure 11 column 4, unlike that of the plain frame channels depicted Figure 11 column 2, are uniform proving the security of the scheme. This demonstration substantiates the robustness of the scheme against any histogram analysis attack.

E. Statistics of the chaos: For an ideally uniformly distributed 8-bit image, the pixel values are expected to be uniformly distributed between 0 and 255, inclusively. Hence, the corresponding ideal reference statistical descriptions are

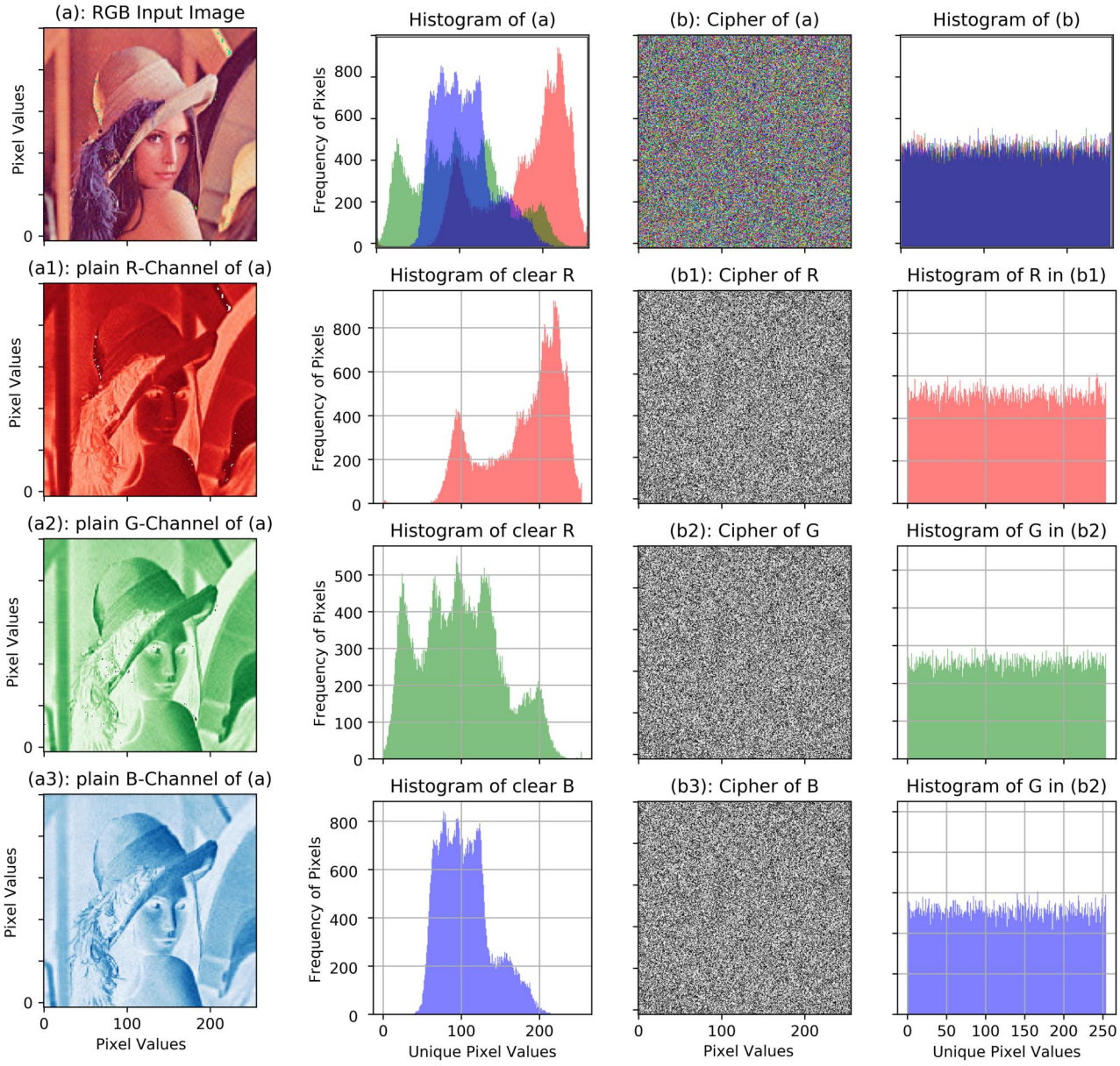

FIGURE 11 Histogram: Column 1 comprises the clear input frame and its colour channels R, G, and B from top to bottom. Column 2 depicts the histograms of the clear input frame and its R, G, and B channels in the order from top to bottom. Column 3 shows the ciphers of the input frame and its colour channels in column 1 in the same order. Column 4 illustrates the histograms of the cipher frame and its channels in column 3

provided in Table 3. A robust scrambling scheme is supposed to produce statistical descriptions close to these ideal values. The cipher of SCM scheme has statistical descriptions, provided in the last row of Table 3, which almost achieve the ideal mean, standard deviation (std), minimum (min), 25th percentile, 50th percentile, 75th percentile, and maximum (max) values. This again validates the uniformity of SCM scheme.

F. Key sensitivity analysis: Key sensitivity analysis measures the difference between two ciphers obtained by enciphering the same plain frame using two slightly different keys. Precisely, a pair of keys that differ only by one bit must produce two entirely different ciphers. The difference between the two ciphers is measured by calculating the NPCR and UACI of the ciphers. The rule of thumb is that these two ciphers must achieve $NPCR > 99\%$ and $UACI$ around 33% in order for the scrambling scheme to be resistant against differential attacks. Defined by Equation (14), UACI is employed to measure the average intensity difference in a colour channel between its two cipher versions $C_1(i, j)$ and $C_2(i, j)$.

TABLE 3 Statistical descriptions

	Mean	std	min	25%	50%	75%	max
Ideal	127.5	73.9	0	63.75	127.5	191.25	255
SCM	127.98	73.55	0	64	128	192	255

Abbreviation: SCM, Sine-cosine Chaotic Map.

TABLE 4 Comparative security and performance analysis

Parameter	SCM	DyCIE	RoI-Mask	2D-ReC	RC4	AES	Liu's	Tang's
Mean	127.43	127.5	127.26	127.49	127.35	127.31	127.89	127.74
std	73.75	73.9	73.78	73.897	73.92	73.93	73.863	73.78
min	0	0	0	0 0	0	0	0	0
25%	63.75	63.75	63.45	63	63	63	64	63
50%	127.45	127.5	128	127	127	127	128	128
75%	191	191.25	191.76	191	191	191	191	192
max	255	255	255	255	255	255	255	255
Key space	2^{405}	2^{320}	2^{512}	2^{448}	2^{2048}	2^{256}	2^{2183}	2^{407}
Key sensitivity								
UACI	0.3352	0.336	0.48	0.3346	0.3345	0.3348	0.3338	0.3337
NPCR	0.9965	0.997	0.99616	0.99673	0.9961	0.9960	0.9966	0.9964
Plain pixels sensitivity								
UACI	0.3312	0.331	0.3543	0.3253	4×10^{-6}	0.3339	0.3342	0.3317
NPCR	0.9958	0.9964	9972	0.9954	0.0001	0.9959	0.9951	0.9961
PSNR (dB)	7.733	10.45	8.73	9.43	7.749	7.740	9.731	9.153
Entropy	7.999	7.9984	7.909	7.998	7.9998	7.999	7.9992	7.999
Horizontal correlation	3.7×10^{-5}	0.0045	8.89×10^{-5}	6×10^{-4}	8.1×10^{-4}	1.4×10^{-3}	4.5×10^{-3}	−0.0485
Vertical correlation	3.3×10^{-5}	0.0072	5.52×10^{-4}	9×10^{-4}	2×10^{-5}	1.5×10^{-3}	0.0039	0.0643
Diagonal correlation	3.6×10^{-3}	0.0025	0.00812	3×10^{-4}	1.2×10^{-3}	4.5×10^{-4}	0.0054	0.0035
fps	10.05	10.274	*N/A*	5.605	3.74	1.024	1.215	0.346

Abbreviations: DyCIE, Dynamic Chaotic Image Enciphering; fps, frames per second; NPCR, Number of Pixels Change Rate; PSNR, Peak Signal to Noise Ratio; RoI-Mask, Regions of Interest Masking; SCM, Sine-cosine Chaotic Map; UACI, Unified Average Changing Intensity.

$$UACI = \frac{1}{H * W}\left[\sum_{i,j}\frac{C_1(i,j) - C_2(i,j)}{255}\right] * 100\% \quad (14)$$

The NPCR measures the change rate of the number of pixels of the cipher-frame when only a bit of the original key or pixel is modified. It is defined by Equation (15).

$$NPCR = \frac{\sum_{i,j} D(i,j)}{H * W} * 100\% \quad (15)$$

where H and W are the height and width of the cipher images, encrypted using key_1 and key_2 that vary from each other by only a bit. $D(i, j)$ is defined by Equation (16).

$$D(i,j) = \begin{cases} 1, & \text{if } C_1(i,j) \neq C_2(i,j) \\ 0, & \text{else} \end{cases} \quad (16)$$

The proposed SCM scheme achieves a $UACI = 33.5\%$ and an $NPCR = 99.69\%$, which means that it meets the key sensitivity requirements and it is secure against possible differential analysis attack. This is also referred to as the differential analysis.

G. Clear-frame pixel sensitivity: Similar to the measure of key sensitivity, the clear-image sensitivity states that a given plain original image and another version with a single bit change should produce completely different ciphers when encrypted using the same key. Our SCM scheme achieves an NPCR of 99.23% and a UACI of 33.12% and is secure against differential attack.

H. Peak Signal to Noise Ratio (PSNR) analysis: PSNR helps us measure how much disparate the original frame and its cipher are. It is defined as the base-10 logarithm of the ratio of the square of the maximum pixel value to the mean square error (MSE) of the plain and enciphered frames. Equation (17) defines the $PSNR$ where W and H are dimensions of a frame (I), C is its scrambled version, and $MAX_I = 255$. The PSNR of a robust scrambling scheme is expected to be low because the MSE of the plain and scrambled images is expected to be higher. The average PSNR of the SCM scheme is 7.733 dB, which is a really great result that signifies huge disparity between the plain frame and its cipher.

$$\begin{aligned} PSRN &= 10 * log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ MSE &= \frac{1}{W * H}\sum_{i=0}^{W-1}\sum_{j=0}^{H-1}[I(i,j) - C(i,j)]^2 \end{aligned} \quad (17)$$

I. Information entropy analysis: The information entropy, $H(C)$, defined by Equation(18), measures how randomly the N pixels of the scrambled image C are shuffled. For an 8-bit pixel representation, the ideal value of the entropy is $H(C) = 8$. Hence, for an 8-bit pixel representation, a secure scrambling scheme is supposed to have an information entropy value very close to 8.0. SCM scheme produces entropy values of 7.999, very close to the ideal value, which proves its security against entropy analysis attack.

$$H(C) = -\sum_{i=0}^{N-1} P(C_i)\log_2(C_i) \tag{18}$$

J. Correlation analysis: Video frames are so bulky characterised by very strong redundancy and correlations amongst adjacent pixels. Hence, to make correlation analysis attack irrelevant, a robust scrambling scheme should be able to diffuse and shuffle frames to produce a cipher with uncorrelated adjacent pixels. The correlation parameter computes the correlation among the adjacent pixels of the cipher frame. A secure frame-scrambling scheme is required to produce a cipher with uncorrelated or nearly zero correlation between adjacent pixels. The correlation analysis in the horizontal, vertical and diagonal directions is performed between pairs of clear-frame channels and cipher-frame channels by using Equation (19).

Our experimental analysis on SCM scheme shows that the correlations of the cipher pixels in the horizontal, vertical, and diagonal directions are nearly zero (0.000037, 0.000033, and 0.0036, respectively). These numbers corroborate the robustness and security of SCM scheme against correlation analysis attacks. More illustratively, Figure 12 (ah), (av), and (ad) portray the horizontal, vertical, and diagonal correlations of the adjacent pixels of the clear input image in Figure 12a, respectively. On the other hand, Figure 12 (bh), (bv), and (bd) illustrate the randomised horizontal, vertical, and diagonal correlations of the adjacent pixels of the cipher in Figure 12b proving its robustness against correlation attacks.

$$\begin{aligned} r_{xy} &= \frac{cov(x,y)}{\sqrt{D(x)D(y)}} \\ cov(x,y) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x_i))(y_i - E(y_i)) \\ E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \end{aligned} \tag{19}$$

9.2 | Dynamic Chaotic Image Enciphering Scheme (DyCIE)

All performance analyses performed on the SCM (Subsection 9.1) in the just-previous subsection were also performed on the DyCIE. It meets all security requirements except it has a narrower range for its control parameters; whereas SCM has an unlimited range. With regard to speed, the proposed DyCIE has similar complexity, $O(n) = R$, and on average, this scheme can scramble about 10.274 fps. All test results of this scheme, along with the results of six other methods are summarised in Table 4.

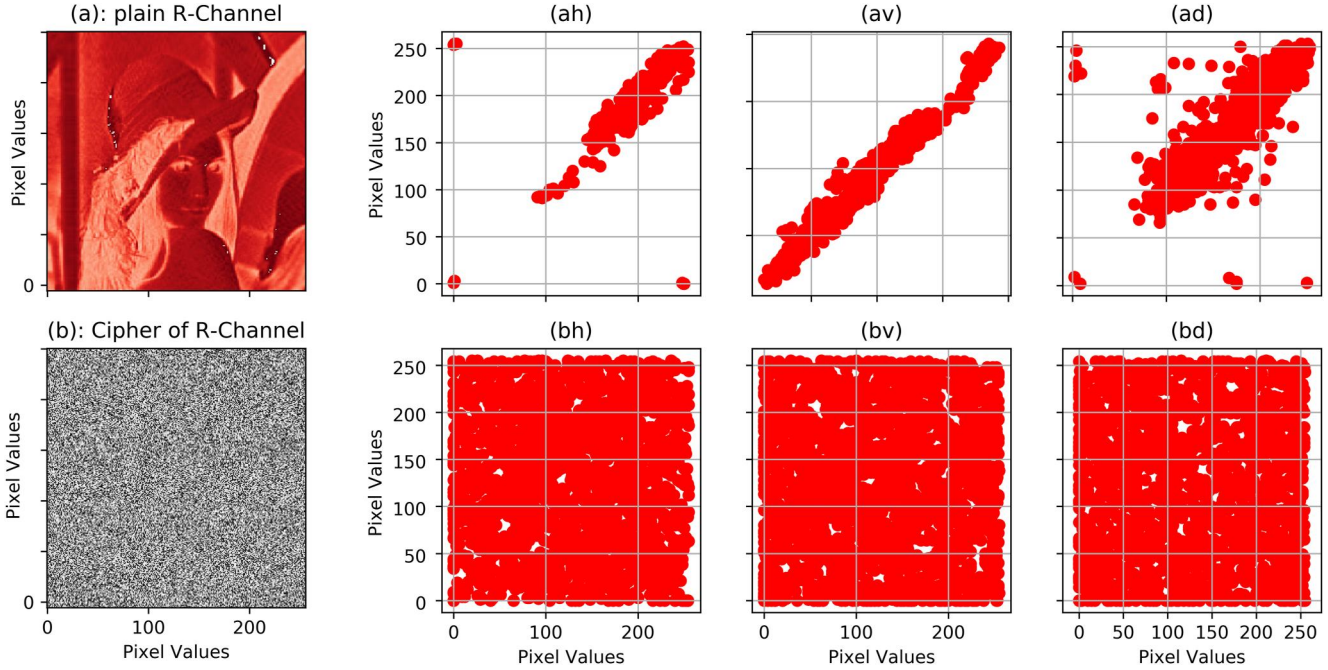

FIGURE 12 An illustration of scatter plots of the horizontal, vertical, and diagonal correlations of the clear R-channel in (a) and its cipher frame in (b)

9.3 | RoI-mask for attributes denaturing

The generated random chaotic sequence passes all standard security tests defined in Subsection 9.1. Figure 13 demonstrates a sample histogram analysis for our RoI-Mask scheme. Figure 13a is a clear boxed input face and its ciphered version is shown in Figure 13b. Figure 13c shows that the frequency distribution of the clear input image in Figure 13a is not uniform. In contrast, the frequency distribution of the scrambled face in Figure 13b has become uniform as portrayed in Figure 13d. This validates that the privacy-protection scheme is secure against a frequency analysis attack. It has a complexity of $O(n) = N$ and is computationally efficient for smaller RoIs. Summing it up, it is proved to be robust against the possible statistical and computational attacks. All the tests carried out are depicted in Table 4.

9.4 | Impacts of block shuffling

Figure 14 presents the experimental results of the frame shuffling process. Figure 14a shows the clear input frame. Then, before the actual scrambling process, the input frame is shuffled in blocks of size 32 × 32 as depicted in Figure 14b. Figure 14c–f show outputs of frame-shuffling with blocks of size of 16 × 16, 8 × 8, 4 × 4, and 1 × 1, respectively. In all of them, there is nothing recognisable. But the degree of pixelation decreases and the degree of confusion increases as the block size decreases. For computational efficacy, we employed a block size of 32 × 32 in this work.

9.5 | Comparative analysis

In addition to a comprehensive security and performance analysis, we have conducted a comparative study not only among our proposals, but also with contemporary chaotic schemes and most widely used data encryption schemes. Our proposed chaotic schemes compared here include SCM, DyCIE, and ROI-Mask. Five benchmarks were selected based on a combination of performance speed, being state of the art, security, and the basic technique employed. They are RC4, AES, an image encryption scheme based on simple logistic chaotic map proposed by Liu et al. [35], an image encryption scheme based on double spiral scans and chaotic maps proposed by Tang

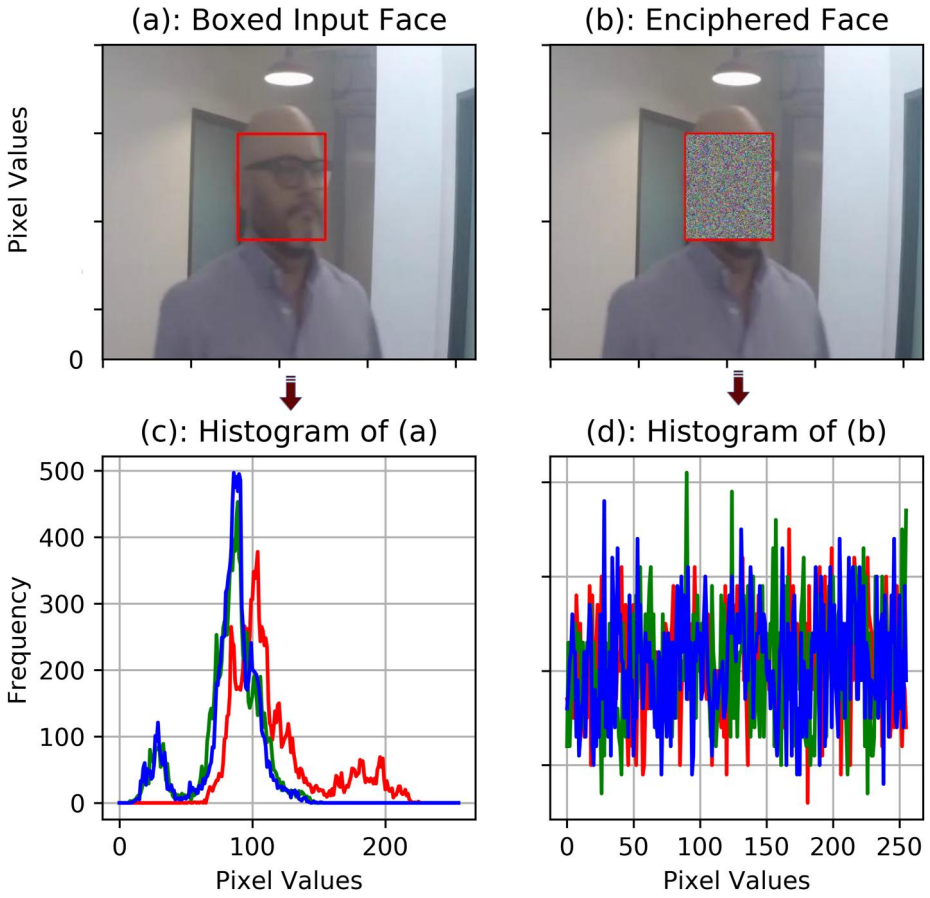

FIGURE 13 Histogram: (a) plain face, (b) cipher of face (a), (c) histogram of clear face, and (d) uniform histogram of cipher face (b)

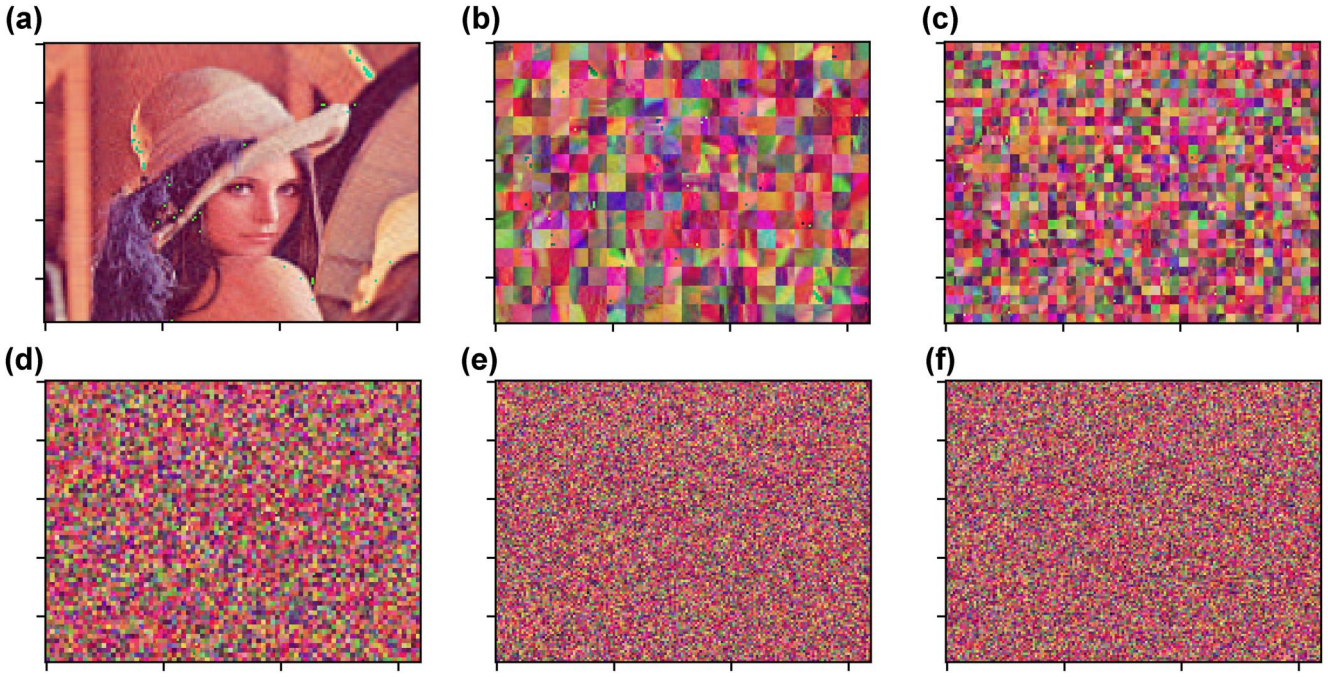

FIGURE 14 Frame shuffling with different block sizes: (a) clear input frame, (b) shuffled with block_size: 32 × 32, (c) shuffled with block_size:16 × 16, (d) shuffled with block_size: 8 × 8, (e) shuffled with block_size: 4 × 4, (f) shuffled with block_size: 1 × 1

et al. [36], and our earlier work, 2D Reproducible Chaotic (2D-ReC) [38]. RC4 is one of the fastest stream ciphers; it has some key security issues, though. AES is the most widely used encryption scheme in today's Internet at the TLS/SSL. Works introduced by Liu et al. and Tang et al. are chaotic-based encryption schemes. The Liu's approach is a relatively lighter scheme designed based on a parameter-varied non-linear logistic chaotic map, the simplest and well-studied chaotic map. It offers good speed and security unlike other higher dimension chaotic systems. The Tang's proposal is one of the most recently published chaotic encryption schemes. It is a secure scheme designed based on low complexity double spiral scans and a chaotic map. The 2D-ReC is a computationally thin scheme for recently proposed full video frame scrambling in a resource-constrained computational environment.

Table 4 summarises the results of the comparative study and analysis. The schemes proposed in this study have achieved very good security and performance. Considering the basic descriptive statistics that measure the uniformity of the pixel distribution of each scheme, the results show that both our proposed schemes and the comparison group have distributions very close to the ideal one. The same is true with the measure of pixel and key sensitivity measures; that is, both our schemes and the benchmarks have comparable results. The PSNR measures how disparate the plain frame and its cipher version are from each other. As clearly portrayed on row 12 of Table 4, the PSNR values of our schemes are smaller proving the randomness of the respective ciphers produced. Furthermore, one front where the traditional encryption schemes like AES do not consistently address in image or frame encryption is the issue of dissociating the strong pixel correlation. It is a known fact that video frames or images comprise bulky information with strongly correlated pixels. Then, our schemes were carefully designed to break any pixel adjacency; as a result, they have better scores in terms of the correlation parameter. Besides, one of the key design goal of this work is making the schemes lightweight that can run on the edge of a network, where there are computing-resource constraints. Hence, our schemes clearly stand out in terms of speed, achieved through better optimisation and design that supports efficient parallel processing. They have way much better frame processing speed than the benchmark methods proving their lightweight design. The SCM, and DyCIE can process about 10.05 fps, and 10.274 fps, respectively. However, the RoI-Mask is not compared with the others in terms of fps because it is specially designed for protecting small RoI on a frame. It takes about 34 ms to scramble a 150 × 150 × 3 RoI on a frame.

Overall, the SCM frame enciphering scheme is superior to the other schemes in terms of security and range of parameter values. It is more robust than the other methods when measured in terms of standard computational and statistical security parameters. It can optimally run on edge cameras where there is resource constraint. The DyCIE scheme works well in a much resource-constrained environment and slightly edges the rest in terms of speed. The RoI-Mask is an ideal candidate for denaturing some RoIs on a video frame.

10 | CONCLUSIONS

Privacy is fundamental to healthy operation of surveillance systems; as a result, it has been among the major concerns while a bunch of smart cities heavily rely on VSS. In this work, we proposed three slenderised mechanisms to preserve E2E privacy in VSS. The design rationales and principles are discussed in detail. A comprehensive performance and security analysis corroborates that the proposed schemes are secure and faster than existing video-frame scrambling methods at the edge of the network. The technology for privacy-aware and privacy-preserving smart cities is far from mature, a lot of open questions yet to be tackled. We hope this effort will inspire more discussions and novel solutions from the edge computing and smart cities community.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Yu Chen https://orcid.org/0000-0003-1880-0586

REFERENCES

1. Chen, N., Chen, Y.: Smart city surveillance at the network edge in the era of IoT: opportunities and challenges. In: Smart Cities, pp. 153–176. (2018)
2. Fitwi, A., et al.: Estimating interpersonal distance and crowd density with a single edge camera. MDPI Computers. 10(11), 143 (2021). https://doi.org/10.3390/computers10110143
3. Mali, D., Hadush, A.: Home monitoring system using wireless sensor network via internet. Technia 7(1), 11014 (2014)
4. Chen, N., et al.: Smart city surveillance in fog computing. In: Advances in Mobile Cloud Computing and Big Data in the 5G Era, pp. 203–226. Springer, Berlin (2017)
5. Fitwi, A.H., et al.: A distributed agent-based framework for a constellation of drones in a military operation. In: 2019 Winter Simulation Conference (WSC), pp. 2548–2559. IEEE (2019)
6. Cavallaro, A.: Privacy in video surveillance [in the spotlight]. IEEE Signal Process. Mag. 2(24), 168–166 (2007)
7. Fitwi, A., Chen, Y.: Privacy-preserving selective video surveillance. In: 2020 29th International Conference on Computer Communications and Networks (ICCCN), pp. 1–10. IEEE (2020)
8. Kumar, V., Svensson, J.: Promoting Social Change and Democracy Through Information Technology. IGI Global, Hershey (2015)
9. Newton, E.M., Sweeney, L., Malin, B.: Preserving privacy by de-identifying face images. IEEE Trans. Knowl. Data Eng. 17(2), 232–243 (2005)
10. Lin, L., Purnell, N.: A world with a billion cameras watching you is just around the corner. Wall Str. J. (2019). https://www.wsj.com/articles/a-billion-surveillance-cameras-forecast-to-be-watching-within-two-years-11575565402
11. Bellovin, S.M.: A look back at security problems in the TCP/IP protocol suite. In: 20th Annual Computer Security Applications Conference, pp. 229–249. IEEE (2004)
12. Fitwi, A., Chen, Y., Zhou, N.: An agent-administrator-based security mechanism for distributed sensors and drones for smart grid monitoring. In: Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII, vol. 11018, p. 110180L. International Society for Optics and Photonics (2019)
13. Fitwi, A., et al.: Smart grids enabled by edge computing (2020)
14. Clement, J.: Global digital population as of July 2020 (2020). Accessed 24 Aug 2020. https://www.statista.com/statistics/617136/digital-population-worldwide/
15. Fitwi, A., et al.: Minor privacy protection by real-time children identification and face scrambling at the edge. EAI Endorsed Trans. Secur. Saf. 18(3), e3 (2020)
16. Feigenbaum, M.J.: The onset spectrum of turbulence. Phys. Lett. A. 74(6), 375–378 (1979)
17. Phatak, S., Rao, S.S.: Logistic map: a possible random-number generator. Phys. Rev. E. 51(4), 3670–3678 (1995)
18. Gleick, J.: Chaos: Making a New Science. Open Road Media, New York (2011)
19. Rakhmawati, L., et al.: Image privacy protection techniques: a survey. In: TENCON 2018-2018 IEEE Region 10 Conference, pp. 0076–0080. IEEE (2018)
20. Padilla-López, J.R., Chaaraoui, A.A., Flórez-Revuelta, F.: Visual privacy protection methods: a survey. Expert Syst. Appl. 42(9), 4177–4195 (2015)
21. Winkler, T., Rinner, B.: Security and privacy protection in visual sensor networks: a survey. ACM Comput. Surv. (CSUR). 47(1), 1–42 (2014)
22. Korshunov, P., Ebrahimi, T.: Using face morphing to protect privacy. In: 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 208–213. IEEE (2013)
23. Ribaric, S., Ariyaeeinia, A., Pavesic, N.: De-identification for privacy protection in multimedia content: a survey. Signal Process. Image Commun. 47, 131–151 (2016)
24. Saini, M., et al.: W 3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video. Multimed. Tools Appl. 68(1), 135–158 (2014)
25. Artusi, A., et al.: Overview and evaluation of the jpeg xt hdr image compression standard. J. Real-Time Image Process. 16(2), 413–428 (2019)
26. Yuan, L., Ebrahimi, T.: Image privacy protection with secure jpeg transmorphing. IET Signal Process. 11(9), 1031–1038 (2017)
27. Pennebaker, W.B., Mitchell, J.L.: JPEG: Still Image Data Compression Standard. Springer Science & Business Media, Berlin (1992)
28. Rajpoot, Q.M., Jensen, C.D.: Security and privacy in video surveillance: requirements and challenges. In: IFIP International Information Security Conference, pp. 169–184. Springer, Berlin (2014)
29. Asim, M., Jeoti, V.: On image encryption: comparison between AES and a novel chaotic encryption scheme. In: International Conference on Signal Processing, Communications and Networking, pp. 65–69. IEEE (2007)
30. Xiao, S., Yu, Z., Deng, Y.: Design and analysis of a novel chaos-based image encryption algorithm via switch control mechanism. In: Security and Communication Networks, vol. 2020 (2020)
31. Fitwi, A.H., Nouh, S.: Performance analysis of chaotic encryption using a shared image as a key. Zede J. 28, 17–29 (2011)
32. Li, R., Liu, Q., Liu, L.: Novel image encryption algorithm based on improved logistic map. IET Image Process. 13(1), 125–134 (2018)
33. Zhang, X., Wang, X.: Chaos-based partial encryption of spiht coded color images. Signal Process. 93(9), 2422–2431 (2013)
34. Zhou, Y., Bao, L., Chen, C.P.: A new 1d chaotic system for image encryption. Signal Process. 97, 172–182 (2014)
35. Liu, L., Miao, S.: A new image encryption algorithm based on logistic chaotic map with varying parameter. SpringerPlus. 5(1), 289 (2016)
36. Tang, Z., et al.: Image encryption with double spiral scans and chaotic maps. In: Security and Communication Networks (2019)
37. Levy, D.: Chaos theory and strategy: theory, application, and managerial implications. Strat. Manag. J. 15(S2), 167–178 (1994)
38. Fitwi, A., et al.: Privacy-preserving surveillance as an edge service based on lightweight video protection schemes using face de-identification and window masking. Electronics 10(3), 236 (2021)
39. Fitwi, A., et al.: Slenderized privacy-preserving surveillance as an edge service. In: 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC), pp. 125–134. IEEE (2020)

40. Nikouei, S.Y., et al.: Decentralized smart surveillance through micro-services platform. In: Sensors and Systems for Space Applications XII, vol. 11017, p. 110170K. International Society for Optics and Photonics (2019)
41. Nikouei, S.Y., et al.: Real-time index authentication for event-oriented surveillance video query using blockchain. In: IEEE International Smart Cities Conference (ISC2), pp. 1–8. IEEE (2018)
42. Xu, R., et al.: Blendmas: A blockchain-enabled decentralized micro-services architecture for smart public safety. In: IEEE International Conference on Blockchain (Blockchain), pp. 564–571. IEEE (2019)
43. Xu, R., et al.: Blendsps: A blockchain-enabled decentralized smart public safety system. Smart Cities 3(3), 928–951 (2020)
44. Yuan, M., et al.: Minor privacy protection through real-time video processing at the edge. arXiv preprint. http://arxiv/org/abs/2005.01178 (2020)

How to cite this article: Fitwi, A., Chen, Y., Zhu, S.: Lightweight frame scrambling mechanisms for end-to-end privacy in edge smart surveillance. IET Smart Cities. 1–19 (2021). https://doi.org/10.1049/smc2.12019