Enforcing Privacy Preservation on Edge Cameras using Lightweight Video Frame Scrambling

Alem Fitwi†, Yu Chen†, Sencun Zhu‡
†Dept. of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA
‡Dept. of Computer Science and Engineering, Penn State University, State College, PA 16802, USA
Emails: {afitwi1, ychen}@binghamton.edu, sxz16@psu.edu

***Abstract*—Protecting privacy is a very challenging task in a highly surveilled world with zillions of surveillance cameras deployed ubiquitously. The difficulty mainly lies in the facts: (i) there is not a distinctively defined boundary between usability and privacy, (ii) video frames indiscriminately created and collected by the edge cameras could be abused and intercepted, and (iii) it is difficult to enforce the commonly used compute-intensive standard techniques as-is on the edge cameras because of limited computational resources. In this paper, we propose a lightweight and secure scheme to Enforce Privacy-preservation on Edge Cameras (EnPEC) using deep learning and a sinusoidal chaotic-map. The proposed EnPEC architecture comprises a lightweight frame classifier designed to label frames as offensive and harmless depending on their content to ensure the practice of selective surveillance following a frame approximation process and a novel sinusoidal-map-based chaotic image scrambling technique that enciphers frames color-channel wise to ensure end-to-end (E2E) privacy of frame contents. The extensive analysis of the functionality, performance and security of the EnPEC scheme, and comparison with related works verify that the EnPEC scheme is more feasible, robust and secure when it runs in real-time on edge cameras equipped with a computational power equivalent to the Raspberry PI 4.**

***Keywords*-Edge Computing, Frame Classification, Privacy, Sinusoidal Chaotic-Map, Video Surveillance.**

I. Introduction

Video surveillance systems (VSS) are widely deployed across the world with the undisguised goal of increasing public security and safety. Closed Circuit Television (CCTV) cameras are the eyes of the video surveillance system where over a billion of them are installed in key and strategic places. However, cameras indiscriminately garner a great deal of visual data containing individuals' personal information, activities, and behavioral patterns without any regard to their privacy. Many incidences of privacy breaches and abuse of videos were reported [13], [19], [41]. As a result, there have been a wide-spread concern of privacy invasion and a persistent call for making the CCTV cameras privacy-aware.

In a CCTV-camera based surveillance system, the signals from the cameras at the edge of the network are analyzed at distant cloud/fog servers after being transmitted over a campus area network (CAN) or wide area network (WAN) and are monitored by security personnel sitting in surveillance operation centers (SOC) away from the cameras. Hence, people in charge of the cameras can abuse them to furtively collect unauthorized information about individuals [10], [13]. In addition, the visual data communicated by the edge cameras to monitors in SOCs could be intercepted while in transit by attackers or leaked by legitimate users and spilled into the wider cyber space causing a lot of damage [9], [10], [14]. Any method proposed to take the privacy issues up, therefore, must have means to remedy the aforementioned problems. However, it is not possible to ensure 100% privacy because it implies zero-usability, a direct opposite to the very purpose of surveillance systems. Hence, a good privacy mechanism is one that can ensure an optimized balance between the privacy and usability of visual data.

The existing dominant client-server architecture of the surveillance system incurs privacy breaches and bandwidth costs [21]. Video frames irrespective of whether they are useful or not are forwarded to distant cloud/fog servers for analytics and then to remote viewing stations causing unnecessary load to the bandwidth. The edge computing paradigm has clear advantages over the cloud/fog computing paradigms in this regard in that it equips edge devices with some computational capability and intelligence important for enforcing privacy measures at the point where the video frames are created [12]. However, edge devices like CCTV cameras have limited computational power. Privacy-preserving mechanisms built based on deep neural network (DNN) or machine learning (ML) and scrambling techniques to classify behavioral patterns and scramble frames to ensure end-to-end (E2E) privacy, respectively, must be designed to be computationally light enough to fit into these resource-constrained devices.

To date, there is not a feasible solution that addresses the glaring privacy breaches by the intrusive practice of surveillance. In this paper, we propose a lightweight and secure scheme to Enforce Privacy-preservation on Edge Cameras (EnPEC) for surveillance services based on deep learning and sinusoidal chaotic-map following a thorough investigation. The contribution of our paper are briefly outlined as follows:

- A lightweight DNN-based frame classifier that identifies three aggressive behaviors, namely knife wielding, gun brandishing, and raising a formed fist, is proposed. It classifies frames into offensive and harmless classes. The frames labeled harmless are dropped to save bandwidth and further processing time and those labeled as aggressive are scrambled and forwarded to the SOC. To improve the frame processing efficiency, we also adopted an efficient frame approximation scheme based on the matrix singular value decomposition (SVD) method.
- A novel Lightweight Sinusoidal Chaotic-map (LiSC)

video-frame scrambling scheme is proposed. It is the kernel of our EnPEC scheme deployed at the edge cameras that ensures E2E privacy of frame contents. LiSC is designed based on solid mathematical analysis and robust security and performance experiments. It enciphers every outgoing frame classified as aggressive by our simplified frame-classifier. LiSC is a secure and comparatively more efficient solution for enciphering frames at edge cameras.

- Extensive experimental and pragmatic security, robustness, and performance analyses are carried out and compared with contemporary techniques. All mathematical and experimental analysis results validate the security, robustness, and good performance of our proposed scheme.

The remainder of the paper is organized as ensues. The related works in the areas of video surveillance practices and privacy challenges, and existing techniques of privacy preservation are presented in Section II. Section III describes the overall architecture of the proposed EnPEC scheme. Section IV presents the frame approximation and frame classification techniques. The novel sinusoidal-chaotic-map based LiSC scheme for video-frame scrambling is expounded in Section V. Section VI presents the detailed experiments and results. Lastly, Section VII concludes the paper.

II. Related Works

A. Video Surveillance Practices and Privacy Challenges

With the unconcealed motive of stopping crimes and ensuring physical security and public safety, the number of CCTV cameras mounted on building walls and corners, perched on street poles, border checkpoints, and lamp posts around the globe has reached a over billion [8]. This enables law enforcement agencies to indiscriminately gather a large amount of visual information about individuals without their knowledge and consent. This practice has caused a lot of brazen privacy breaches even though privacy is enshrined in many constitutional documents as the fundamental human right and the state of being free from being disturbed or observed by other people without one's consent [3], [26], [39]. Consequently, there have been public outcries and at times outrages calling concerned bodies to make the surveillance practice privacy-aware. One cause of the concern is the fact that the cameras lack such behavior-discriminating intelligence and collect data of innocent individuals engendering their privacy through interception, misuse or deliberate leak.

There are a number of reports and publications that confirm the invasion of individual's privacy by the invasive practice of CCTV surveillance. For instance, the American Civil Liberties Union (ACLU) has identified five abuses of CCTV cameras, which are criminal abuse, institutional abuse, abuse for personal purposes, discriminatory targeting, and voyeurism [30]. Targeting and spying political opponents using CCTV cameras to silence them is another abuse. In addition, the higher degree of maneuverability that today's cameras offer is abused to direct cameras to furtively capture individuals via openings (windows) while doing private stuffs at home. One notable high-profile spying incidence by means of CCTV cameras is that of Angela Merkel, who was spied by means of a museum's camera operated by a security guard [11]. By this virtue and due to possible interception of videos while in transit, VSS endangers the privacy of individuals. Huge visual data about individuals could be either abused or divulged into the wider cyber space. Another challenge is balancing out the usability and privacy. Focusing on specific privacy attributes like the face might not be sufficient to preserve privacy. Still individuals can be identified by means of other attributes like their gait, clothing, and other distinguishing visual marks in the frames [10], [13]; hence, an optimal balancing is needed.

B. Video Privacy Protection Techniques

Almost all of the efforts that have been made so far focus mainly on masking some privacy attributes like face and are cloud-based protections [6], [10], [32], [36], [41], [40]. Since 2012, there have existed a number of convoluted neural networks (CNN) like VGG-Net [31], Res-Net [18], and many others [4], [33]. They were developed based on deeply involved and computationally expensive networks. As a result, they are more suitable for cloud computing paradigms. The Facebook's DeepFace [34] and Google's FaceNet [29] achieved significant improvements over traditional approaches and yield near-human accuracy today. However, they cannot fit into the edge devices owning to computing-resource constraints. They cannot process even one frame per second (fps) at the edge. Hence, the trend of research has recently changed from highly convoluted networks to lightweight ones; the lightweight ones cannot process more than 4fps on an edge device like Raspberry Pi 4, though. A number of research outputs have attempted to leverage this advancement to localize privacy-sensitive objects. Most of the pertinent works focus on recommending social media users about the privacy-setting they should enforce before sharing an image to their social network pages [1], [36], [40]. These works give some insight about the attributes deemed private by many individuals but they cannot address the privacy problems in surveillance systems. They do not provide solutions for addressing E2E privacy problems.

With regards to video scrambling, there are many methods available today ranging from face regions, false color, JPEG to editing schemes [26]. The face regions approach [23], [27] suffers from not being suitable to real-time processing and lack of reversibility. In a similar fashion, the false color [5] and JPEG [2] methods also suffer from similar problems. Besides, apart from encryption, most of the editing schemes are unable to completely hide sensitive contents on images. They are also prone to reconstruction attacks [24], [38]. They include simple schemes like blurring, black box, pixelation, and masking. But most importantly, information is not recoverable. Looking into the encryption schemes, public-key cryptographic schemes like Rivest, Shamir and Adleman (RSA) and Elliptical Curve Cryptography (ECC) are too slow to be used for bulky data encryption. Besides, the traditional symmetric key cryptographic mechanisms like the malleable and fast Rivest Cipher 4 (RC4) and Advanced Data Encryption

(AES) are not convenient for video enciphering. The RC4 is a very fast stream cipher but it is not considered as a secure cipher any longer due to its vulnerability to a bit-flipping attack. The AES is the most widely used block cipher in today's transport layer security (TLS) of the Internet. However, its real-time video encryption speed on an edge-device and its ability to break the strong correlation of adjacent pixels of an image are not impressive. Hence, chaotic image-encryption techniques are the best solutions due to their better performance, high degree of sensitivity to slight changes in initial conditions, higher degree of randomness, enormous key space, and high security. Nowadays, there are many robustly secure chaotic systems out there proposed by a number of researchers [11], [15], [20], [25], [35], [42]. However, they are still compute-intensive and slow to be deployed in a resource-constrained environment, like the edge of a network. Only lighter versions of them can make sense on edge devices.

C. Survey Summary

Putting our survey findings in a nutshell, most of the related works are cloud-based solutions [1], [36] and do not address the issues of E2E-privacy. There also other works that focus on specific security attributes [40] including our previous works [10], [11]. However, they don't address the problems of E2E-privacy and usability-privacy balance problem. With respect to video encryption mechanisms, well, the public-cryptographic schemes are too slow to be used at the edge. The RC4 is fast enough to meaningfully encipher frames on edge devices but it suffers from key bit-flipping attacks. AES is the most widely used and most secure symmetrical-encryption scheme in today's Internet; nonetheless, it is slow for an edge computing setting and unable to fully break strong associations among adjacent pixels. Generally, chaotic-based solutions are the most suitable ones for video enciphering due to their higher randomness, security, and speed. Nevertheless, high dimensional (2D, 3D, or more) chaotic schemes are not suitable for edge environment because they are too slow. Only computationally-thin chaotic solutions can be meaningfully applied at the edge. For example, previously, we designed and developed a highly secure and lightweight 2D chaotic-solutions based on the complex solutions of a homogeneous second-order differential equation [11]; however, its computational speed is only about 50% of what we achieved in this work (EnPEC). Furthermore, we developed a secure chaotic scheme based on the De Jong Map for the scrambling of region of interests (ROIs) on frames [13]; however, that cannot be employed for ensuring E2E privacy of video frames.

Therefore, in this paper, we have tried to address the privacy challenges by introducing a lightweight frame classifier and scrambling scheme that ensure the privacy of individuals on video frames content-wise at the edge of the network. It ensures E2E privacy without undercutting the benefits of VSS. That is, behavioral patterns are identified by means of deep-learning model and outgoing frames of interest are scrambled using a computationally-thin chaotic-based scrambling scheme at the point of video creation. In contrast to the previous works, our EnPEC scheme alleviates the privacy problems by using a secure and lighter scheme that can meaningfully run at the edge of the network. Putting it another way, a novel lightweight sinusoidal-chaotic-map video-frame scrambling scheme, whose performance and security were comprehensively analyzed, is proposed for secure and more efficient frame scrambling on edge cameras. It was designed and implemented with careful considerations of the flaws suffered by previous related works.

III. Architecture of EnPEC Scheme

In the preexisting VSS, tremendous amount of information about individuals moving along urban streets, visiting stores or offices is garnered by using zillions of edge cameras mounted on walls and ceilings or perched on poles. These huge amount of data are often processed and analyzed by humans sitting in distant SOCs aided by some centralized software and hardware-based video-processing tools. This practice often invades the privacy of individuals in that the indiscriminately collected data could be misused or intercepted by adversaries while in transit from the edge of the network to the SOCs. Hence, creating lightweight smart agents that can perform frame classification and frame scrambling at the edge cameras is of huge importance in the process of protecting the privacy of individuals. Figure 1 illustrates the proposed EnPEC architecture, which is an edge-based, privacy-preserving smart surveillance system where the privacy measures are enforced at the edge cameras, the network nodes at which videos are created. At the edge, three major tasks, namely frame approximation, frame classification, and frame scrambling are performed to ensure the privacy of individuals caught on cameras. At the distant SOC, viewing of selected offensive frames for follow-up actions and frame storage for the purpose of later use are performed after the video frames have been transmitted on the network and unscrambled, as described in Algorithm 2.

1. Frame Approximation: Edge cameras are resource constrained. Hence, removing some of the redundant information from every frame without drastically compromising the quality is vital in improving the computational complexity on edge devices installed at the edge of the network. Hence, the frame-approximation is performed immediately after the creation of frames.

2. Frame Classification: Privacy is a basic human right found enshrined in many constitutions. Therefore, a trade-off between privacy and value is of paramount importance. In EnPEC, frames containing offensive or violent behaviors like knife wielding, gun brandishing, swinging a punch are labeled as "offensive" frames. Those frames that contain no vicious or violent behavioral patterns are labeled as "harmless". The compromise made is that the privacy of individuals in harmless frames is fully protected whereas the privacy of individuals involved in offensive acts is partly protected.

3. Frame Scrambling: One way the privacy of individuals caught on video frames is undercut is through interception by intruders during transmission over a network from the point of creation to the distant analytics centers. Hence, the best

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

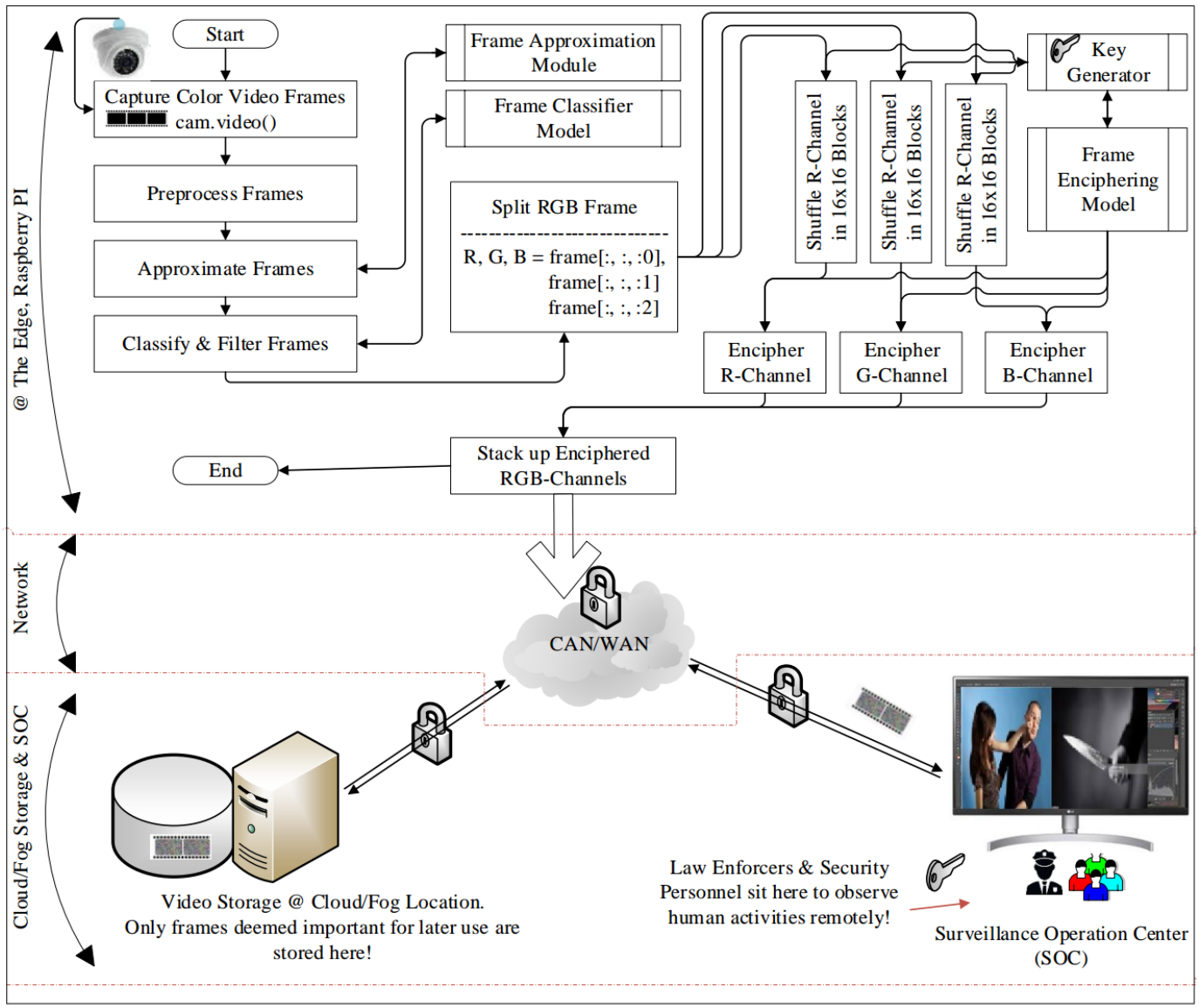


Fig. 1. Overall EnPEC System Architecture: Edge cameras, frame approximation module, frame-classifier model, and frame-scrambling module.

solution is to encrypt frames using a thin video encryption scheme. The frame scrambling module comprises such sub-modules as key generator and frame-scrambler designed based on a lightweight but secure sinusoidal chaotic map. It encrypts only those frames that need to be transported to the servers or SOC following classification by the frame-classifying model. **4. Frame Viewing and Storage**: The frames, containing information of interest to security personnel, are transported to the viewing and storage centers in the distant locations as portrayed in Fig. 1. This enables the law enforcement people to take prompt actions and use the stored footage as evidence in courts of laws against the law-breakers.

IV. Video Frame Classification

It is essential to add intelligence to the edge cameras so as to have the capability to discriminate offensive and harmless scenarios. For privacy reasons, the harmless frames are dropped, and only the ones with offensive contents are encrypted and sent to the viewing and storage centers over the Internet or a CAN. Before the classification process, frame approximation is performed to improve the processing speed without significantly affecting the quality of video frames.

A. Frame Approximation

To reduce the computational complexity at the edge cameras where there are resource-constraints, we employed an appropriate frame dimensionality reduction achieved via frame decomposition to obtain low-rank approximations of the matrices of the frame pixels. The decomposition is done by performing pseudo-inverses of non-square matrices to find the solution of a system of equations stated in Eq. (1).

$$Ax = b \tag{1}$$

The high-dimensional input frames are decomposed into their most statistically descriptive factors. The image pixel matrix approximation is performed using the Eckart-Young theorem [16], which states that the optimal rank r

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

approximation to frame I, in a least-squares sense, is given by the rank r singular value decomposition (SVD) truncation $\tilde{I}$ in Eq. (2). The rank, r, refers to the number of singular values.

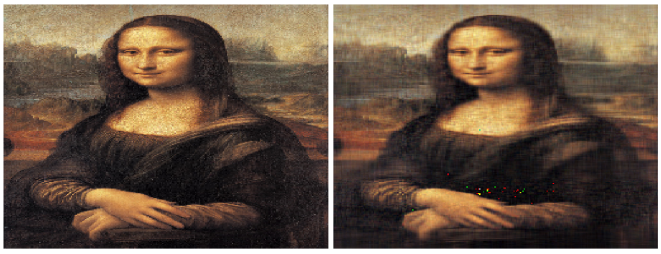
$$argmin \ = \ ||I - \tilde{I}||_F, \ \tilde{I} \ s.t. \ rank(\tilde{I}) = r \qquad (2)$$

The SVD is performed on every pixel-matrix of every frame channel to compress it by removing redundant information. Eq. (3) SVD decomposes a rectangular matrix M of a frame to three parts.

$$M \ = \ U\Sigma V^T \qquad (3)$$

where U is a matrix of left singular vectors in the columns, Σ is a diagonal matrix with singular values and V is a matrix of right singular vectors in the columns.

A sample output of the approximation process is illustrated in Fig. 2. An input frame/image of size 480P portrayed in Fig. 2(a) is pre-processed using the SVD scheme and its memory size is reduced by about 25.4% without drastically affecting the quality of the image. The approximated image is presented in Fig. 2(b). This saves memory and bandwidth. This is inline with the customary practice of performing data compression before encryption for encryption randomizes the data rendering compression inefficient.



(a) Original Input Image (b) Approximated Image

Fig. 2. Frame Approximation Using an SVD Method.

B. Frame Classifier Model

The frame classifier model was built based on the CNN illustrated in Fig. 3 with Keras. It comprises convolutional layers (Conv2D) that perform a set of mathematical operations to produce a single value in the output feature map. The second important component of the network is the Pooling layers, which down-sample the image data extracted by the convolutional layers to reduce the dimensionality of the feature map so as to decrease the processing time. The most widely used pooling algorithm is the max pooling (MaxPooling2D), which extracts sub-regions of the feature map based on its size, which keeps the maximum value while discarding all other values. The third important component of the network is the dense (fully connected) layers. It performs classification on the features extracted by the convolutional layers and down-sampled by the pooling layers. In a dense layer, every node is connected to every node in the preceding layer as depicted in Fig. 3. Summing it up, we developed this model after thoroughly working on the data, model construction, objective function, and optimization phases in an iterative manner.

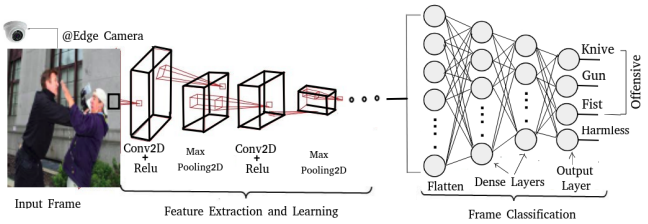


Fig. 3. Simplified Frame Classifier Model.

The frame classifier model comprises six Conv2D layers and two Dense layers. The kernel size employed is 3×3. The other parameters used include MaxPooling2D of size 2×2, Rectified Linear Unit (ReLU) activation function for the Conv2D layers and first dense layer, and a softmax activation function for the output Dense layer. The typical number of filters used are 32, 64, 128, and 256 in increasing order. It was trained based on pre-processed 40,000 images collected from various sources including EgoGesture, nvGesture, Kaggle, soft computing, Edgecase.ai, and the dataset published by Grega [17]. To consider real scenarios where frames could be caught in various angles, the 40,000 images were further augmented through flipping, zooming, and shear-range variation.

V. Preserving Privacy Using Sinusoidal Maps

Image encryption is of paramount importance for ensuring E2E protection of privacy in a VSS. However, the traditional cryptographic mechanisms like RC4, AES, RSA, and ECC are not quite suitable for real-time video frames/images encryption because of the slow computational speed. Chaotic encryption schemes have very high sensitivity to slight changes on initial conditions and parameters. These important attributes make chaotic systems the better choices for image encryption. However, these schemes as well suffer from high computational complexity. The existing ones are not suitable for deployment on edge cameras which have very limited computational resources.

Being inspired by the the logistic map [7], [22], [25], [37], a well-known low-dimension chaotic system, we proposed a novel Lightweight Sinusoidal Chaotic-map (LiSC) for video frame encryption at the edge of the network. As stated in Eq. (4), the variables α, β and γ are the control parameters, and x_0 is the initial condition for the proposed LiSC scheme.

$$x_{n+1} \ = \ \frac{\alpha}{2}[2sin(\beta x_n) + \gamma(1 - cos(2\beta x_n))] \qquad (4)$$

With multiple control parameters, the computationally-thin LiSC scheme has a wider interval for some of the control variables, better chaos performance with more uniform distribution of points, much larger key space, and better security. LiSC passes all the standard security tests described in subsection VI-B. The LiSC system has a double curve unlike the simple logistic map, which has a single downward parabolic curve.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

$$\begin{aligned}
[x_0,\ \alpha,\ \beta,\ \gamma,\ \delta,\ \theta] &= gen_key() \\
x_n &= x_0 \\
tmp_1 &= \alpha \times sin(\beta \times x_n) \\
tmp_2 &= 0.5 \times \alpha\gamma cos(2\beta \times x_n) \\
const &= 0.5\alpha\gamma \\
x_{n+1} &= 1624[tmp_1 - tmp_2 + const]\%1 \\
Chaotic_sequence &= append(x_{n+1} \times \theta) \\
x_n &= \delta \times x_{n+1}
\end{aligned} \tag{5}$$

After a thorough security and statistical analysis, a multiplying constant was added to Eq. (4) to further enhance the security of the chaotic sequence generator. It generates a secure and evenly distributed chaotic image. Eq. (5) describes how the key and chaos are generated, with three more parameters added to those stated in Eq. (4). They are a multiplier 1624, a scaling parameter θ and an updating parameter δ. 1624 was obtained experimentally.

In EnPEC, the video frame encryption using the LiSC sinusoidal chaotic sequence is performed image color-channel-wise simultaneously to improve the processing time. The image scrambling process comprises a key-generator and chaos-generator modules. The key generator module (key_gen()) generates a key comprising six elements, each 64-bit long, as stated in Eq. (6).

$$Key = [x_0,\ \alpha,\ \beta,\ \gamma,\ \delta,\ \theta] = genKey() \tag{6}$$

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 color channels of the image are generated by using Eq. (7 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 between two values (a, b), where there are infinitely many real-valued numbers between them.

$$\begin{aligned}
Key_R &= [r_0,\ \alpha_r,\ \beta_r,\ \gamma_r,\ \delta_r,\ \theta_r] = genKey() \\
Key_G &= [g_0,\ \alpha_g,\ \beta_g,\ \gamma_g,\ \delta_g,\ \theta_g] = genKey() \\
Key_B &= [b_0,\ \alpha_b,\ \beta_b,\ \gamma_b,\ \delta_b,\ \theta_b] = genKey()
\end{aligned} \tag{7}$$

To enable a quick encryption of the color channels of every frame, the required chaotic images ($Chaos_r$, $Chaos_g$, and $Chaos_b$) are produced in parallel. Eq. (8) illustrates the calculation of chaotic image for channel Red ($Chaos_r$). The chaotic images of channel Green $Chaos_g$ and Blue $Chaos_b$ are generated similarly.

$$\begin{aligned}
t_r &= (1 + \gamma_r \times sin(\beta_r \times r_0)) \\
R_c &= 1624 \times \alpha_r \times sin(\beta_r \times r_0) \times t_r\%1 \\
r_0 &= R_c \times \delta_r \\
Chaos_r &= R_c \times \theta_r
\end{aligned} \tag{8}$$

To further enhance the security of the EnPEC scheme, a simple but efficient shuffling algorithm is introduced. It randomizes the pixels of a frame in blocks of sizes $1 \times 1,\ 2 \times 2,\ 4 \times 4,\ 8 \times 8,\ 16 \times 16,\ or\ 32 \times 32$. This increases the diffusion and confusion of pixels which makes the scheme highly robust and secure against any differential attacks. The procedure is depicted in Algorithm 1 pythonically. Packages and modules like r_, numpy (np), pandas (pd), random, and itertools (it) are imported from python during implementation. In other words, as portrayed in Fig. 5 in Subsection VI-B of Section VI, the input frame's pixels are shuffled randomly just like the shuffling of playing cards in blocks of size 16×16.

Algorithm 1 Frame-Pixels Shuffling

```
W, H ← 640, 480
procedure SHUFFLE_FRAME(frame, blk_size)
    h ← [i for i in range(0, H, blk_size)]
    w ← [j for j in range(0, W, blk_size)]
    hw ← list(it.product(h, w))
    dfhw ← pd.DataFrame(hw, columns = ['h','w'])
    dfhw['i'] ← lst
    dfhw ← dfhw.sample(frac = 1)
    hws ← list(zip(dfhw['h'].tolist(), dfhw['w'].tolist()))
    index_key ← dfhw.index.tolist()
    hws ← np.asarray(hws)
    hws ← hws.reshape(int(H/blk_size), int(W/blk_size))
    imsize ← ch.shape
    imgn ← ch
    c1 ← 0
    for i in r_[:imsize[0]:blk_size] do
        c2=0
        for j in r_[:imsize[0]:blk_size] do
            x, y = hws[c1][c2]
            imgn[i:(i + blk_size),j:(j+
blk_size)] =
                ch[x:(x + blk_size),y:(y +
blk_size)]
        c1+=1
    return ch, index_key
```

As depicted in Algorithm 1, all possible block positions (x, y) are combined together as tuples in a list. Then, they are truly shuffled after an index has been added. The shuffling method employed is inherently irreversible; however, the index is used to reproduce the original tuple positions. The block size, and $index\_key$ employed to shuffle the input frame at the sender must be forwarded to the receiver as part of the key. At the receiver, the original positions of the blocks in the frame are restored by simply sorting the $index\_key$ column.

Algorithm 2 describes the whole tasks performed by the privacy-preserving EnPEC scheme. It first approximates the frame by removing unnecessarily redundant components or information using the SVD method. Then, a frame classification is performed by using the frame classifier model followed by the scrambling process. At last, frames deemed important are forwarded to remote server, operation centers, and storage sites in scrambled form.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

Algorithm 2 EnPEC: Privacy-preserving Scheme

@ Sending End (Edge Camera)
np := numpy for faster and vectorized processing
$(knife, gun, fist, harmless) \leftarrow (0, 1, 2, 3)$
vid $\leftarrow$ videoCapture()
W, H $\leftarrow$ *640, 480*
w, h $\leftarrow$ *256, 256*
procedure APPROX_FRAME(frame)
    $f\_approx \leftarrow SVD(frame)$
    return f_approx
procedure CLASSIFY_FRAME(f_{approx})
    $f_{approx} \leftarrow f_{approx}$*.resize((w, h))*
    $result \leftarrow frame\_classifier\_model(f_{approx})$
    if $result\ in\ [0, 1, 2]$ **then**
        return offensive
    else
        return harmless
procedure GEN_KEY
    $key \leftarrow Eq.7$
    return key
procedure GEN_CHAOS(key)
    $chaos \leftarrow Eq.8$
    return chaos
procedure SCRAMBLE_FRAME(frame, chaos)
    $frame \leftarrow shuffle\_frame(frame)$
    $frame_{enc} \leftarrow frame \oplus chaos$
    return frame$_{enc}$
procedure UNSCRAMBLE_FRAME($frame_{enc}$, chaos)
    $frame_{enc} \leftarrow unshuffle(frame_{enc})$
    $frame_{clear} \leftarrow frame_{enc} \oplus chaos$
    return frame$_{clear}$
while True **do**
    status, frame $\leftarrow$ vid.read()
    frame $\leftarrow$ *frame.resize((W, H))*
    if *status* **then**
        Channels are approximated in parallel
        $f_{approx} \leftarrow approx\_frame(frame)$
        if f_{type} == *'offensive'* **then**
            Channel keys are generated in parallel
            $key_r, key_g, key_b \leftarrow gen\_key()$
            $key \leftarrow key_r, key_g, key_b$
            Channel chaoses are generated in parallel
            $chaos \leftarrow gen\_chaos(key)$
            $chaos_r, chaos_g, chaos_b \leftarrow chaos$
            Color channels are scrambled in parallel
            $f_{enc} \leftarrow scramble\_frame(f_{approx}, chaos)$
        else
            continue
@ Receiving End (Server, SOC, or Storage sites)
while True **do**
    $f_{enc}, key \leftarrow from\ sender\ (camera)$
    $chaos \leftarrow gen\_chaos(key)$
    Channel decryptions in parallel
    $f_{clear} \leftarrow unscramble\_frame(f_{enc}, chaos)$

VI. Experimental Studies and Analysis

A comprehensive experimental study of the proposed lightweight EnPEC scheme has been conducted. The experiments and analysis of EnPEC scheme have been implemented in Python and C++. Then, they are tested in real-time on edge nodes equipped Raspberry Pi 4, onto which all modules are loaded. For all experiments and analyses, a standard RGB-color frame of size 480P ($480 \times 640 \times 3$) was considered, where the scrambling is processed color-channel wise in a multi-threaded and multi-tasked manner. For the inference process by the frame classifier, input frames are resized to $256 \times 256 \times 3$ for the model was trained based on such input size to reduce the time complexity.

A. Frame Classification

The created model is lightweight with a size of 15.6MB that successfully discriminates the frames into harmless and offensive with an average accuracy of 96.04%. Figure 4 demonstrates how the classification is performed. Three of the input frames ((a), (b), and (c)) are offensive and frame (d) is harmless. The model has correctly labeled the frames as can be seen from Fig. 4 output frames. In other words, a frame predicted as 0, 1 and/or 2 is labeled as "offensive" while a frame predicted as 3 is labeled as "harmless".

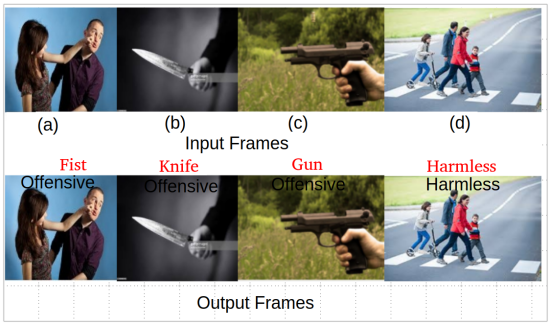


Fig. 4. Frame classification: Fist raising in frame (a) is predicted 98.5%, knife in frame (b) is predicted with 100%, gun in frame (c) is predicted 100%, and harmless behavioral pattern in frame (d) is predicted 97.35%

.

The frame discrimination process, on top of improving the privacy, improves the bandwidth and storage utilization. Let us consider a video comprising 100 frames, out of which only 17 contains aggressive behavioral patterns and 83 of them are harmless. In the absence of a frame discriminator at the edge, 100 of the frames will be forwarded to the distant analytics servers and SOCs for viewing and storage. But with the frame discriminator enforced at the edge, only 17 of the frames will be forwarded saving 83% of the bandwidth and storage usage.

B. Scrambling Offensive Frames

For the comprehensive performance and security analysis of the proposed LiSC scheme, considerable number of cases and parameters have been employed. The properties, parameters, and security analysis considered include functional tests, Lyapunov Exponents, time complexity, visual assessment, key space, key and pixel sensitivities, statistical analysis, Peak

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

Signal to Noise Ratio (PSNR), Number of Pixels Change Rate (NPCR), Unified Average Changing Intensity (UACI), histogram and correlation analysis, and Information Entropy analysis.

1) Frame Shuffling Test: Figure 5(a) shows the clear input frame. Before the scrambling process, the input frame is divided into blocks of size 32×32 and shuffled as depicted in Fig. 5(b). Figures 5 (c), (d), (e), (f), and (g) show the outputs of shuffling with blocks of sizes 16, 8×8, 4×4, 2×2, and 1×1, respectively. In all of them, there is nothing recognizable. 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 16×16 in this work.

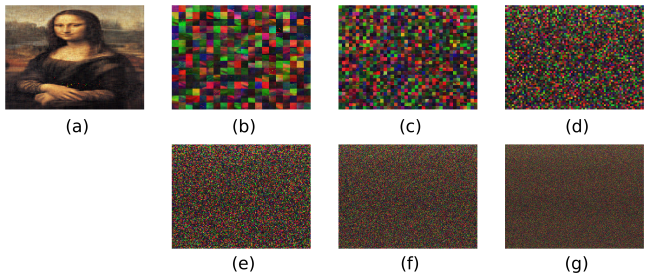


Fig. 5. Frame shuffling with different block sizes: (a) Clear input frame, (b) shuffled with block_size: 32×32, (c) shuffled with block_size:16×16, (d) shuffled with block_size: 8×8, (e) shuffled with block_size: 4×4, (f) shuffled with block_size: 2×2, and (g) shuffled with block_size: 1×1

2) Overall Functional Test: As depicted in Fig. 6, the input frames predicted as offensive by the frame classifier were successfully scrambled by the scrambling module and forwarded while the frame labeled as harmless was dropped. Figure 6 also validates that the proposed scrambling scheme functions as designed.

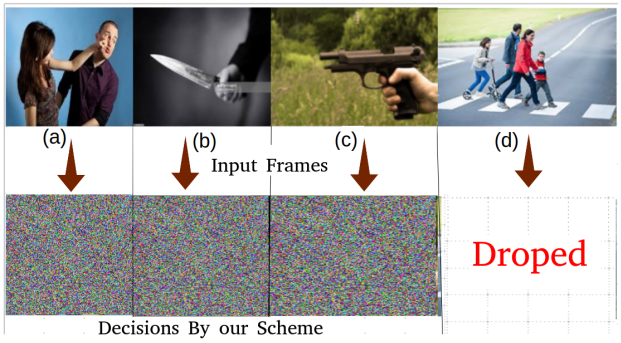


Fig. 6. Frame scrambling: Aggressive frames (a), (b), & (c) are enciphered & sent while a harmless frame (d) is dropped.

3) 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 can be considered as 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 Eq. (9). The Lyapunov exponent indicates how even the points of a chaotic sequence generator are distributed, as computed by Eq. (10).

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

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

The Lyapunov exponents of our proposed LiSC method is computed as follows based on Eq. (10).

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

where $t = 0.5 \times \alpha \times \gamma$. Then, the derivatives on Eq. (11) were performed by the application of calculus rules like product rule $((f(x)g(x))' = f'(x)g'(x))$ and chain rule $((f(g(x))' = (fog)' = f'(g(x)) * g'(x))$.

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

Hence, the final formula for computing the Lyapunov exponents of our LiSC scheme is provided in Eq. (13), obtained by substituting $\frac{dx_{n+1}}{dx_n}$ in Eq. (10) with its expression given by Eq. (12). The Lyapunov Diagram of the proposed LiSC is generated by Eq. (13) and shown in Fig. 7.

$$LE = \lim_{N \to +\infty} \frac{1}{N} \sum_{n=1}^{N} log|\alpha\beta[t_1 + t_2]| \tag{13}$$

where $t_1 = cos(\beta \times x_n)$ and $t_2 = \gamma \times \sin(2\beta \times x_n)$.

Figure 7 illustrates the Lyapunov Diagram of our LiSC scheme. The figure demonstrates that Lyapunov exponents of the sinusoidal map are more evenly distributed over unrestricted range. For illustration purpose, the control variable is fixed at 40 but it can be extended to any greater value. The Lyapunov exponents vary within a very narrow range (4.374 and 4.437) signifying the uniformity of our LiSC scheme, which is way much better than other low dimension chaotic methods, like the popular Logistic map [25] as shown in Fig. 8. The logistic map in Fig. 8 vary between -4 and 1 indicating non-uniformity, and has uneven Lyapunov properties and experiences chaotic properties only within a restricted range of the control variable, between 3.57 and 4.

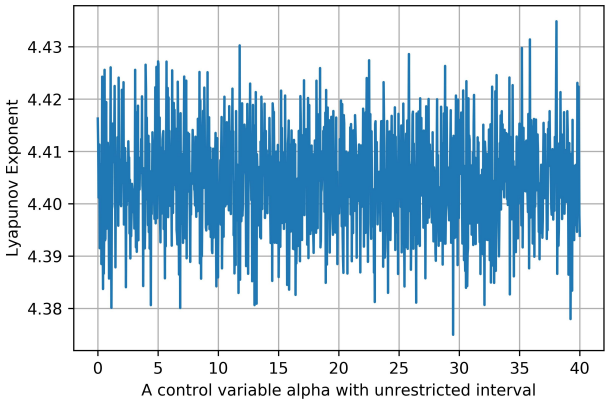


Fig. 7. Magnified Lyapunov Diagram of the proposed LiSC scheme.

4) Encryption Time: Video frames are bulky, and a good-performing scrambling technique is one that takes less time to encrypt a frame. The computational speed depends upon

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

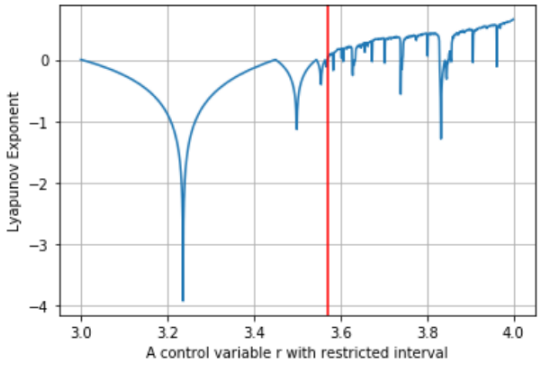


Fig. 8. Lyapunov Diagram of the Logistic Map [25].

the type of hardware platform, and the programming language employed. In this work, Raspberry PI 4 is adopted as the edge device and all results presented are based on it. Besides, we implemented prototypes using both Python 3.7.4 and $C++$. As expected, the implementation based on $C++$ is much faster than the python implementation. However, for the purpose of comparative analysis with equivalent previous works, the python implementation is consistently considered. Our LiSC scheme, therefore, can encrypt about 10.5 frames per second (fps), which is sufficient for real-time safety surveillance considering the velocity of pedestrians (often $\leq 8m/s$).

5) Key Space Analysis: In the common practice of cryptography, the security depends on the size of the key space. It should be sufficiently large and unbreakable by brute force analysis in reasonably short period of time. Our LiSC scheme has a key that comprises six parameters, each 64-bit long floating-point decimal value. Hence, the total key space of our scheme is 2^{384}, which is sufficiently large to resist any possible exhaustive key search analysis attack.

6) Visual Assessment: A scrambling scheme is said to be good if there is no recognizable visual information in the cipher image. As illustrated in Fig. 9, the scrambled frame (Fig. 9(b)) contains no visually recognizable information about the clear frame (Fig. 9(a)). The encrypted image is random-like and highly disordered proving that the scheme is resistant against any visual assessment attack.

7) Histogram Analysis: As the frequency description of each pixel value of a frame, the histograms of the scrambled frame must be totally statistically different from that of corresponding original images. The histograms of the scrambled versions shown in Fig. 9 (b1) - (b3), unlike that of the plain frame channels depicted in Fig. 9 (a1) - (a3), are uniform. This substantiates the robustness of the LiSC scheme against any histogram analysis attacks.

8) Statistics of the Chaos: For an ideally uniformly distributed 8-bit image, the pixel values are uniformly distributed between 0 and 255, inclusive. The corresponding ideal reference statistical descriptions are provided in the first row of Table I. A good scrambling scheme is supposed to produce statistical descriptions close to these ideal values. The statistics of the clear frame in Fig. 9(a) in the second row of Table I is quite different from the ideal statistics. But the cipher of LiSC scheme has statistical descriptions shown in the bottom row of Table I, which are almost equal to the ideal values. This again validates the uniformity of our LiSC scheme.

TABLE I
STATISTICAL DESCRIPTIONS OF THE PROPOSED SCHEME

	mean	**std**	**min**	25%	50%	75%	**max**
Ideal	127.5	73.9	0	63.75	127.5	191.25	255
Clear	132.87	22.89	72	113	135	153	177
LiSC	127.64	74.13	0	63.72	127.83	191.67	255

9) Key Sensitivity Analysis: It measures the difference between two ciphers obtained by enciphering the same plain frame using two slightly different keys. The requirement is that 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 the ciphers produced from the same frame encrypted using an original key and another key with a single bit change from the original key must achieve $NPCR > 99\%$ and $UACI$ around 33% in order for the scrambling scheme to be resistant against differential attacks. The UACI is defined by Eq. (14), which is employed to measure the average intensity difference in a color channel between its two cipher versions $C_1(i,j)$ and $C_2(i,j)$.

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

The NPCR, defined by Eq. (15), 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.

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

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 Eq. (16).

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

With a UACI of 33.85% and an NPCR of 99.73%, our LiSC scheme meets the key sensitivity requirements and it is secure against possible differential analysis attack.

10) Clear-image Pixel Sensitivity: Similar to the measure of key sensitivity, the clear-image sensitivity states that a given plain original image and its another version with a single bit change should produce completely different ciphers when encrypted using the same key. Similarly, NPCR and UACI are employed to compute the difference between the ciphers. Our LiSC scheme produces an NPCR of 99.67% and a UACI of 33.54% proving its security against differential attacks.

11) Peak Signal to Noise Ratio (PSNR) Analysis: It measures the difference between the clear frame and its cipher. 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. Given a plain $W \times H$ 2D color component of an RGB frame, I, and its noisy/scrambled

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

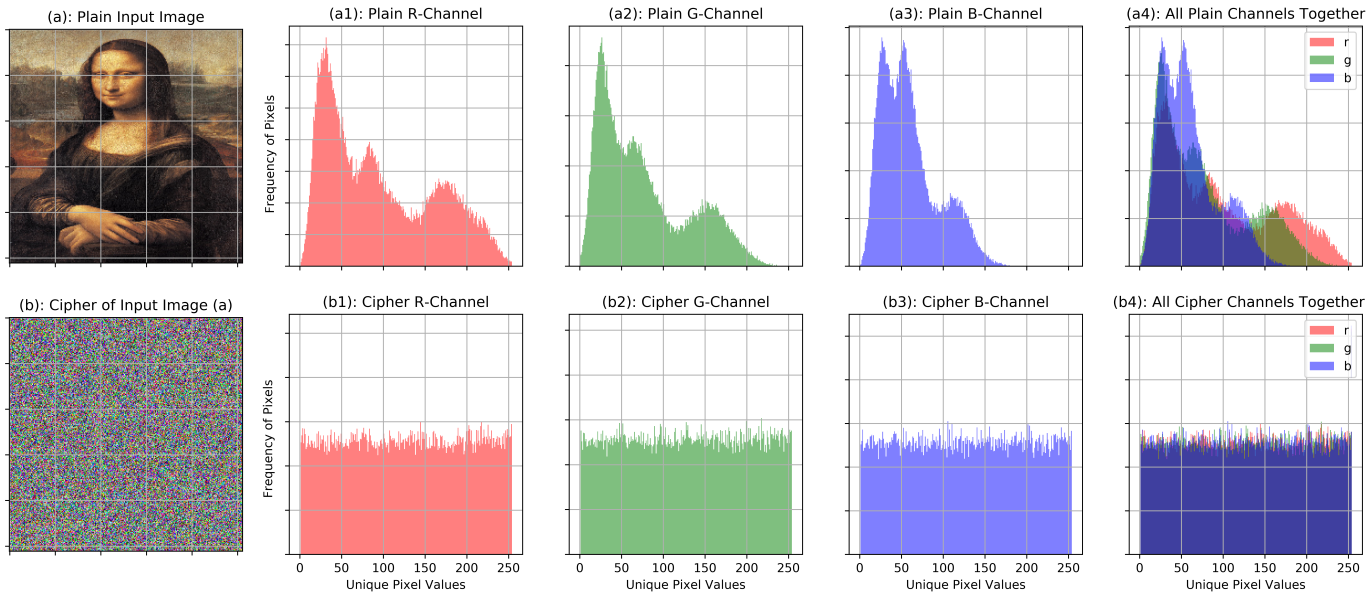


Fig. 9. Visual Assessment: (a) is a plain input image, (a1) - (a3) are the non-uniform histograms of R, G, and B channels of (a); (b) is cipher of (a) and (b1) - (b3) are the uniform histograms of the R, G, and B channels of (b). The histograms of all channels of the plain and cipher images are assembled together on (a4) and (b4), respectively. On the figure, r, g, and b stand for red, green, and blue colors, respectively.

version C, the PSNR is defined as ensues by Eq. (17):

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

MAX_I is the maximum possible pixel value of the plain frame, which is 255 when the pixels are represented using 8 bits per sample. The PSNR of a good 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 proposed LiSC scheme is $7.42\ dB$, which signifies the complete difference between the plain and cipher frames.

12) Information Entropy Analysis: The information entropy, $H(C)$, defined by Eq.(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 $H(C) = 8$. Our LiSC scheme produces an entropy value $H(C) = 7.999$, proving its security against entropy analysis attacks.

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

13) Correlation Analysis: Video frames contain bulky information characterized by very strong redundancy and correlations amongst adjacent pixels. Hence, a secure frame-scrambling scheme is supposed to completely dissociate the correlations of adjacent pixels. Horizontal, vertical and diagonal correlation analyses are performed between pairs of plain-frame and cipher-frame channels using Eq. (19).

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

Our LiSC scheme has achieved nearly zero correlations of the enciphered frame pixels in the horizontal (0.00037), vertical (0.00033), and diagonal (0.00029) directions. This result corroborates the robustness and security of our LiSC scheme against correlation analysis attacks. More illustratively, Figs. 10(c), (e), and (g), respectively, portray the almost linear horizontal, vertical, and diagonal correlations of the adjacent pixels of the plain input image in Fig. 10(a). Meanwhile, Figs. 10(d), (f), and (h) illustrate the random horizontal, vertical, and diagonal correlations of the adjacent pixels of the cipher in Fig. 10(b).

14) Comparative Analysis: After carrying out comprehensive security and performance analysis on our proposal, we have also compared it with equivalent contemporary and most widely used data encryption schemes using the same parameters on the same computing environment. Four benchmark methods are selected based on their good speed, their being state-of-the-art, their good security, the basic techniques employed, or a combination of these criteria. They are RC4, AES, Liu's image encryption scheme based on simple logistic chaotic map [20], and Tang's image encryption scheme based on double spiral scans and chaotic maps [35]. The RC4 is one of the simplest and fastest stream

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3135352, IEEE Transactions on Services Computing

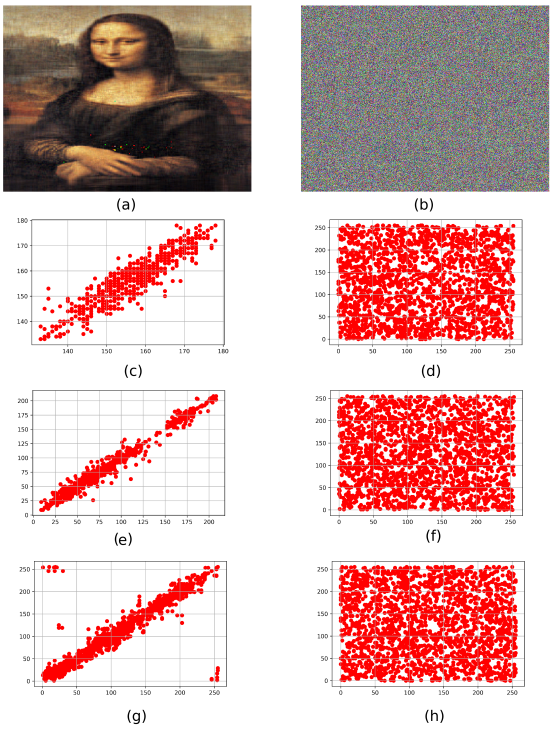


Fig. 10. Scatter plots: (a) plain input frame; (c) horizontal correlation of clear frame (a); (e) vertical correlation of clear frame (a); (g) diagonal correlation of clear frame (a); (b) encrypted version of the input frame (a); (d) horizontal correlation of cipher (b); (f) vertical correlation of cipher (b); and (h) diagonal correlation of cipher (b).

ciphers; however, it has some key security issues. AES is the most widely used block cipher encryption standard in today's Internet TLS/SSL. The works of Liu and Tang are chaotic-based encryption schemes. The Liu's 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 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.

Table II illustrates the statistical, security, and computational performances of our proposed scheme compared with four other methods. For starters, the table portrays that both our LiSC scheme and the comparison group have similar descriptive statistics whose values are very close to the ideal one. Besides, our scheme and the benchmarks have comparable results in terms of key and pixel sensitivity measures; our scheme has a slight edge, though. Considering the PSNR, correlations, and FPS parameters, our LiSC scheme outperforms all other schemes. This proves that our scheme produces a cipher completely disparate from the clear frame and it effectively dissociates the adjacent pixels of an image/a video frame. Furthermore, our scheme was carefully designed to be lightweight; as a result, it is much faster than all other methods at the edge of a network. All in all, the LiSC frame encryption scheme is faster, more robust and secure than the other methods. It can encrypt more than 10 fps on edge cameras where there are acute resource constraints.

TABLE II
COMPARATIVE SECURITY AND PERFORMANCE ANALYSIS

Parameter	LiSC	RC4	AES	Liu's	Tang's
mean	127.53	127.35	173.31	127.89	127.74
std	73.43	73.92	73.93	73.863	73.78
min	0	0	0	0	0
25%	63	63	63	64	63
50%	127	127	127	128	128
75%	191	191	191	191	192
max	255	255	255	255	255
Key Space	2^{384}	2^{2048}	2^{256}	2^{2183}	2^{407}
Key Sensitivity					
UACI	0.3385	0.3345	0.3348	0.3338	0.3337
NPCR	0.9973	0.9961	0.9960	99.66	99.64
Plain Pixels Sensitivity					
UACI	0.3354	4E-6	0.3339	0.3342	0.3317
NPCR	0.9967	0.0001	0.9959	0.9951	0.9961
PSNR	7.42	7.749	7.740	9.731	9.153
Entropy	7.999	7.9998	7.999	7.9992	7.999
Horizontal Correlation	2.9E-5	8.1E-4	0.00139	0.0045	-0.0485
Vertical Correlation	3.1E-5	2E-5	0.00149	0.0039	0.0643
Diagonal Correlation	2.92E-4	1.23E-3	4.5E-4	0.0054	0.0035
FPS	10.469	3.74	1.024	1.215	0.346

C. Summary of Analysis

In summary, the proposed lightweight frame classifier model has a size of 15.6 MB and it can classify the approximated frames into harmless and offensive with an average accuracy of 96.04%. The proposed sinusoidal-chaotic-map based frame scrambling scheme, LiSC, has excellent security, statistical, and computation performances. Particularly, it has higher randomness, smallest cipher-pixel correlations and much better processing speed. It can process about 10.5 fps on an edge-camera. Overall, implemented on an edge camera, the prototype of the EnPEC system including the frame classifier and the scrambler achieved a speed of 5.28 fps, which is better than other lightweight DNN schemes. For instance, MobileNet-Tiny can process 4.5 fps on a Raspberry Pi [28]. Besides, we have implemented the MobileNetv2 on a Raspberry Pi 4, and it achieved below four fps.

VII. CONCLUDING REMARKS

This paper presents EnPEC, a hierarchical and multi-tasked architecture for real-time video frames approximation, classification and scrambling in order to enable a privacy-protecting edge surveillance service. The proposed architecture provides efficient utilization of resources in a multi-threaded, multi-processed, and vectorized manner. It labels frames as harmless or offensive with an average accuracy of 96.04%

providing maximum privacy protection to contents on harmless frames by dropping them while providing partial protection to contents on offensive frames. The scrambling scheme, LiSC, enciphers frames labeled offensive at 10.5 fps ensuring that they can only be accessed by authorized personnel. The whole proposed schemes put together can process 5.28 fps on a camera fitted with a single Raspberry PI 4. In addition, all standard security analyses performed prove its robustness and security against possible attacks. In comparison with existing schemes, it has a higher frame processing rate, highest randomness, and excellent security.

REFERENCES

[1] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49–58, 2019.

[2] A. Artusi, R. K. Mantiuk, T. Richter, P. Hanhart, P. Korshunov, M. Agostinelli, A. Ten, and T. Ebrahimi, "Overview and evaluation of the jpeg xt hdr image compression standard," *Journal of Real-Time Image Processing*, vol. 16, no. 2, pp. 413–428, 2019.

[3] W. C. Bennett, *Civilian drones, privacy, and the federal-state balance*. Center for Technology Innovation at Brookings Washington DC, 2014.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[5] S. Çiftçi, P. Korshunov, A. O. Akyüz, and T. Ebrahimi, "Using false colors to protect visual privacy of sensitive content," in *Human Vision and Electronic Imaging Xx*, vol. 9394. International Society for Optics and Photonics, 2015, p. 93941L.

[6] F. Dufaux, "Video scrambling for privacy protection in video surveillance: recent results and validation framework," in *Mobile Multimedia/Image Processing, Security, and Applications 2011*, vol. 8063. International Society for Optics and Photonics, 2011, p. 806302.

[7] M. J. Feigenbaum, "The onset spectrum of turbulence," *Physics Letters A*, vol. 74, no. 6, pp. 375–378, 1979.

[8] A. Fitwi and Y. Chen, "Secure and privacy-preserving stored surveillance video sharing atop permissioned blockchain," *arXiv preprint arXiv:2104.05617*, 2021.

[9] A. Fitwi, Y. Chen, and N. Zhou, "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. International Society for Optics and Photonics, 2019, p. 110180L.

[10] A. Fitwi, Y. Chen, and S. Zhu, "No peeking through my windows: Conserving privacy in personal drones," *arXiv preprint arXiv:1908.09935*, 2019.

[11] A. Fitwi, Y. Chen, S. Zhu, E. Blasch, and G. Chen, "Privacy-preserving surveillance as an edge service based on lightweight video protection schemes using face de-identification and window masking," *Electronics*, vol. 10, no. 3, p. 236, 2021.

[12] A. Fitwi, Z. Yang, Y. Chen, and X. Lin, "Smart grids enabled by edge computing," 2020.

[13] A. Fitwi, M. Yuan, S. Y. Nikouei, and Y. Chen, "Minor privacy protection by real-time children identification and face scrambling at the edge," *EAI Endorsed Transactions on Security and Safety: Online First*, 5 2020.

[14] A. H. Fitwi, D. Nagothu, Y. Chen, and E. Blasch, "A distributed agent-based framework for a constellation of drones in a military operation," in *2019 Winter Simulation Conference (WSC)*. IEEE, 2019, pp. 2548–2559.

[15] A. H. Fitwi and S. Nouh, "Performance analysis of chaotic encryption using a shared image as a key," *Zede Journal*, vol. 28, pp. 17–29, 2011.

[16] G. H. Golub, A. Hoffman, and G. W. Stewart, "A generalization of the eckart-young-mirsky matrix approximation theorem," *Linear Algebra and its applications*, vol. 88, pp. 317–327, 1987.

[17] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a cctv image," *Sensors*, vol. 16, no. 1, p. 47, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] V. Kumar and J. Svensson, *Promoting social change and democracy through information technology*. IGI Global, 2015.

[20] L. Liu and S. Miao, "A new image encryption algorithm based on logistic chaotic map with varying parameter," *SpringerPlus*, vol. 5, no. 1, p. 289, 2016.

[21] D. Mali and A. Hadush, "Home monitoring system using wireless sensor network via internet," *Technia*, vol. 7, no. 1, p. 11014, 2014.

[22] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[23] E. M. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," *IEEE transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, 2005.

[24] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "Visual privacy protection methods: A survey," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4177–4195, 2015.

[25] S. Phatak and S. S. Rao, "Logistic map: A possible random-number generator," *Physical review E*, vol. 51, no. 4, p. 3670, 1995.

[26] L. Rakhmawati *et al.*, "Image privacy protection techniques: A survey," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 0076–0080.

[27] S. Ribaric, A. Ariyaeeinia, and N. Pavesic, "De-identification for privacy protection in multimedia content: A survey," *Signal Processing: Image Communication*, vol. 47, pp. 131–151, 2016.

[28] N. S. Sanjay and A. Ahmadinia, "Mobilenet-tiny: A deep neural network-based real-time object detection for rasberry pi," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 647–652.

[29] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[30] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 50–57, 2005.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] C. Streiffer, A. Srivastava, V. Orlikowski, Y. Velasco, V. Martin, N. Raval, A. Machanavajjhala, and L. P. Cox, "eprivateeye: To the edge and beyond!" in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. ACM, 2017, p. 18.

[33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[34] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[35] Z. Tang, Y. Yang, S. Xu, C. Yu, and X. Zhang, "Image encryption with double spiral scans and chaotic maps," *Security and Communication Networks*, vol. 2019, 2019.

[36] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, "Enabling live video analytics with a scalable and privacy-aware framework," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 3s, p. 64, 2018.

[37] E. W. Weisstein, "Feigenbaum constant," *delta*, vol. 5, p. 6, 2003.

[38] T. Winkler and B. Rinner, "Security and privacy protection in visual sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 1–42, 2014.

[39] R. Xu, S. Y. Nikouei, D. Nagothu, A. Fitwi, and Y. Chen, "Blendsps: A blockchain-enabled decentralized smart public safety system," *Smart Cities*, vol. 3, no. 3, pp. 928–951, 2020.

[40] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1005–1016, 2017.

[41] M. Yuan, S. Y. Nikouei, A. Fitwi, Y. Chen, and Y. Dong, "Minor privacy protection through real-time video processing at the edge," *arXiv preprint arXiv:2005.01178*, 2020.

[42] Y. Zhou, L. Bao, and C. P. Chen, "A new 1d chaotic system for image encryption," *Signal processing*, vol. 97, pp. 172–182, 2014.