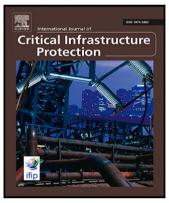
Contents lists available at ScienceDirect

International Journal of Critical Infrastructure Protection

journal homepage: www.elsevier.com/locate/ijcip



SuPOR: A lightweight stream cipher for confidentiality and attack-resilient visual data security in IoT[☆]

Ifeoluwapo Aribilola [a,b,c,*], Saeed Hamood Alsamhi [a,d,e], John G. Breslin [a,b], Mamoona Naveed Asghar [c]

[a] Insight SFI Research Centre for Data Analytics, Data Science Institute, University of Galway, University Road, Galway, H91 AEX4, Co Galway, Ireland
[b] School of Engineering, College of Engineering, University of Galway, University Road, Galway, H91 TK33, Co Galway, Ireland
[c] School of Computer Science, College of Science and Engineering, University of Galway, University Road, Galway, H91 TK33, Co Galway, Ireland
[d] Department of Computer Science and Engineering, College of Informatics, Korea University, 145 Anamro, Seongbukgu, Seoul, Republic of Korea
[e] Electrical Engineering Department, Faculty of Engineering, IBB University, Ibb, Yemen

ARTICLE INFO

Keywords:
One-time key
Galois field
Lightweight stream cipher
Möbius transformations
Non-linearity
S-box
IoT
Attack resilient

ABSTRACT

The rapid growth of Internet of Things (IoT) technologies, particularly visual sensors such as cameras and drones, has resulted in increased transmission of sensitive visual data containing personally identifiable information (PII). Securing this data during storage and transmission (e.g., cloud or edge servers) is essential for maintaining privacy and security. However, existing encryption methods often face challenges due to computational overhead and vulnerability to attacks, especially on resource-limited IoT devices. To bridge this research gap, this paper presents *SuPOR*, a single-round lightweight cipher tailored for visual data protection in IoT environments. The *SuPOR* framework incorporates five fundamental cryptographic principles—**Su**bstitution, **P**ermutation, **XOR**, right circular shift, and swap—which are executed in sequential steps. These include: (1) constructing a secure S-box using Möbius linear transformations and Galois fields for pixel-level substitution, (2) permuting the substituted pixels to improve diffusion, (3) applying a cryptographically secure pseudo-random number generator (CSPRNG) to generate a 64-bit one-time key for **XOR**ing, (4) performing right circular shifts on pixel byte arrays, and (5) executing element swaps to further obfuscate the data. Comprehensive security and statistical assessments demonstrate that *SuPOR* offers strong resistance against various attack vectors while maintaining minimal computational overhead, with a linear time complexity of $O(nm+n(3\times framesize))$. Experimental comparisons indicate that *SuPOR* surpasses several state-of-the-art stream ciphers designed for IoT visual data, making it highly suitable for real-time, resource-constrained environments. The findings provide a practical and efficient solution to enhance the privacy and security of visual data in IoT systems, effectively safeguarding sensitive information from threats.

1. Introduction

Security is considered a critical aspect of the fast-growing Internet of Things (IoT) technologies due to their continuous collection and exchange of data over the Internet [1]. IoT technologies are widely used in wearable devices, smart monitoring systems (cities/homes/buildings/vehicles), industrial control systems, health systems [2], etc. These devices collect real-time data to improve decision-making, maintain the safety of homes and public buildings, and also to improve customer services [3].

Constrained IoT surveillance devices have limited resources to operate, such as memory, processing power, and energy. These devices are designed to perform functions with limited capabilities, making them lightweight and low cost but also limiting their ability to handle critical

☆ This publication has emanated from research supported in part by the European Digital Innovation Hub Data2Sustain, co-funded by Ireland's National Recovery and Resilience Plan (the EU's Recovery and Resilience Facility), the Digital Europe Programme, and the Government of Ireland, and by a grant from Taighde Éireann - Research Ireland under Grant Numbers $12/RC/2289\_P2$ (Insight) and $21/FFP-A/9174$ (SustAIn). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

* Corresponding author at: Insight SFI Research Centre for Data Analytics, Data Science Institute, University of Galway, University Road, Galway, H91 AEX4, Co Galway, Ireland.
E-mail addresses: ifeoluwapo.aribilola@insight-centre.org (I. Aribilola), saeed.alsamhi@insight-centre.org (S.H. Alsamhi), john.breslin@universityofgalway.ie (J.G. Breslin), mamoona.asghar@universityofgalway.ie (M.N. Asghar).

https://doi.org/10.1016/j.ijcip.2025.100786
Received 21 May 2025; Received in revised form 12 July 2025; Accepted 16 July 2025
Available online 25 July 2025

security measures. The increasing prevalence of such devices in our daily lives calls for security measures that can operate efficiently and effectively within their constraints. Real-time data collection, including Personal Identifiable Information (PII), whether in text or visual form (captured through cameras, dashboard cameras, and unmanned aerial vehicles (UAV)), is usually transmitted and stored by third parties (cloud storage), which are often vulnerable to hacking and cyber attacks [4]. To protect recorded information, the General Data Protection Regulation (GDPR) recommends "encryption" as the only reversible data protection solution for all forms of personal data [5]. Encryption can be applied to multimodal data [6–11] to achieve mandatory CIA triad-assisted security services [5]:

1. **Confidentiality (C):** Information that is sensitive or classified can only be restricted to or viewed by certain authorised individuals or systems [12].
2. **Integrity (I):** Ensures that the data are accurate and consistent before being accessed by the appropriate individuals.
3. **Availability (A):** Data remains available in an unmodified form to end users. The availability of data should not suffer from any attack.

For providing security, existing state-of-the-art (SOTA) ciphers (e.g., Advanced Encryption Standard (AES)) are computationally expensive due to their complex and repeated rounds of encryption [13]. With these devices' low voltage and short battery life, complex ciphers significantly reduce their operation time by rapidly draining the batteries. To address the security and computational challenges associated with protecting stored and captured visual data by IoT devices, the following research questions (RQs) are addressed in this paper:

RQ1: What type of cryptographic principles are mandatory to design a reliable lightweight stream cipher to secure visual data (in full or partial form) on constrained IoT devices?
RQ2: How to measure the security robustness of the designed lightweight stream cipher against vulnerable differential attacks, key modification attacks, slide attacks, brute-force attacks, and chosen-plaintext attacks?

1.1. Motivations and contributions

Strong encryption is required to safeguard sensitive personally identifiable information during real-time data transfer to edge/cloud servers due to the growth of visual surveillance IoT devices. However, current stream ciphers are either too computationally demanding for IoT devices with limited resources or are susceptible to chosen-plaintext, brute-force, and differential attacks. By combining substitution-permutation-XOR-shift-swap operations into a single-round lightweight stream cipher, the proposed *SuPOR* (**Su**bstitution-**P**ermutation-X**OR**-Circular_shift-Swap) framework fills the gap. *SuPOR* optimises for linear time complexity, has low memory overhead, and has demonstrable attack resistance, guaranteeing compliance, effectiveness, and security for dynamic visual data in IoT ecosystems. Furthermore, in response to the aforementioned RQs, the paper contributions are summarised as follows:

- **Lightweight Stream Cipher**: This paper proposes a lightweight stream cipher *SuPOR* for securing IoT-captured visual data using five cryptographic principles in a single round.
- **Security Analysis**: We validate SuPOR robustness against differential attacks and brute-force/key-guessing attacks and achieve faster encryption with minimal memory overhead. For GDPR compliance, we propose an S-box and masked visual data based on the Möbius transformation that supports both naïve and selective encryption modes for dynamic visual surveillance.
- **Performance Evaluation**: *SuPOR* is extensively tested through different analyses such as security analysis, statistical analysis, computational cost analysis and comparative analysis with existing security approaches. Usually, stream ciphers are designed and tested for partial/selective encryption. However, *SuPOR* is evaluated for both naïve and selective encryption on real-time videos.

1.2. Paper structure

The remainder of this paper is organised as follows; Section 2 describes the related pieces of literature on cryptography and the substitution box (S-box). Section 3 presents the methodology for designing the lightweight cipher (*SuPOR*) along with the design of the S-box, evaluation, pseudocode, and computational complexity. Section 4 outlines the attributes of the SOTA stream ciphers with their operations. Section 5 discussed the dataset and test-bed configuration specifications and elaborated on the results of implementing *SuPOR*, security, statistical, and computational cost analysis of *SuPOR* in comparison with SOTA ciphers, also with limitations and future work. The conclusion is discussed in Section 6.

2. Background and literature survey

This section reviews the background and current studies on encryption, ciphers, and the substitution box.

2.1. Types of encryption and ciphers

Cryptography describes both encryption (securing) and decryption (retrieving) processes for data protection. As discussed previously, encryption is a reversible way of encoding data to an unreadable format so that only approved individuals can decode it. Encryption can be performed as Naïve (or Full) encryption (NE) and selective encryption (SE) [14]. In NE, the video is fully encrypted but uses more memory and computational time [15], while in SE, the important areas in the video are encrypted, like moving objects [11], faces [16], motion, and texture with less memory and computation. The SE usually implements a detection algorithm such as the Gaussian mixture method (GMM) [17], optical flow (OF) [18], advanced flow of motion detection (AFOM) [19], and others that will select the essential area in the video.

Encryption is categorised into asymmetric and symmetric types. Asymmetric encryption uses a public key for encryption and a private key for decryption [20], while symmetric encryption uses a single key for both encryption and decryption [21]. Symmetric encryption is faster than asymmetric encryption [22], making symmetric encryption ideal for IoT devices. In addition, symmetric encryption is classified into block ciphers and stream ciphers with key lengths, and generating mechanisms determine their strength [21].

In the stream cipher, the pixel digits are encrypted by applying time-varying transformations to the video data. The stream cipher uses the confusion principle, the XOR operative, and converts data into one byte (8 bits) at a time before encryption. However, the block cipher encrypts fixed-size pixel digits as a block using both diffusion and confusion principles. Usually, blocks are divided into octaves (64-bit or 128-bit) [23]. Different block and stream ciphers for IoT with their limitations are discussed in [24,25]. A brief overview of existing SOTA ciphers is given below.

Daniel Bernstein [26] created Chacha20, a stream cipher that performs four quarter rounds of encryption using an integer addition module, bitwise exclusive-or (XOR), and n-bit left rotation each 20 times. This gives a total of 80 quarter-rounds with a 256 bits key length. Salsa20, a stream cipher was also developed by Daniel Bernstein [27] with a 256-bit key length in 20 rounds of operation. Salsa20 uses 32-bit addition, 32-bit XOR, and constant distance of 32-bit rotation. Salsa20 was found to be highly vulnerable to the differential power analysis

(DPA) attack. The authors [28] discussed Rivest Cipher (RC4) as a stream cipher developed by Ron Rivest. It uses a key size of 40–2048 bits. RC4 runs very quickly in software and was considered secure until the BEAST attack exposed its vulnerabilities.

The blowfish block cipher [29] was designed with a 64-bit block size, 16 rounds of encryption, 32-bit to 448-bit keys varying in sizes with 4 S-boxes. Blowfish is vulnerable to plain-text attacks and birthday attacks. PRESENT [30] is a lightweight block cipher with a block size of 64-bit and 80-bit or 128-bit keys. 64-bit block ciphers are prone to block collisions, and a truncated differential attack affected 26rounds of PRESENT. The industry standard cipher Advanced Encryption Standard (AES) [13] is a block cipher that implements 128, 192, 256 bits key lengths and 10, 12, and 16 rounds, respectively. Each round includes sub-byte substitution, shiftrows, mixcolumns and add round key. One of the AES encryption modes is Ciphertext Feedback (CFB) which can be implemented as a stream cipher. AES is NIST (National Institute of Standards and Technology) [31] approved and [22,32] recommended AES for video encryption. However, AES has complex rounds of operation and has proven to be computationally expensive for battery-operated devices.

All discussed ciphers have limitations with respect to security or efficiency for IoT devices. Considering it, this paper proposes an efficient and robust cipher "*SuPOR*" and its effectiveness for IoT cameras is proved by comparing it with the SOTA (AES-CFB, Chacha20, Salsa20, and XOR) ciphers.

2.2. Substitution box (S-box)

Substitution (bytes/pixels) is the strongest security principle of robust ciphers. S-boxes play a pivotal role in creating confusion and are designed to ensure that the relationship between the plaintext and the ciphertext is as complex and non-linear as possible. This makes it difficult for attackers to learn any information about the plaintext from the ciphertext, or to make any progress in breaking the cipher through brute-force attacks. The literature shows the variety of implementations of S-boxes by researchers.

The authors [33] designed a dynamic and key-dependent S-box using a linear trigonometric transformation. The nonlinearity analysis of the S-box gave a minimum of 110 and a maximum of 112, with an average strict avalanche criterion (SAC) of 0.506. This algorithm was applied to images and not videos. The authors [34] used a 3D plasma system that split into several independent chaotic attractors to design an S-box. The result of the nonlinearity is a minimum of 104 and a maximum of 110, while the SAC value is an average of 0.4978. This research did not implement the S-box with an encryption algorithm. The study [35] implemented S-box using a new compound chaotic system method. This method has a minimum nonlinearity of 104 and a maximum of 110. The average result of SAC was 0.49937. The authors [36] proposed three different types of S-box based on genetic algorithm. The first S-box had the best nonlinearity value of 105 minimum and 110 maximum and an average of 0.5197 for the SAC. The study [37] designed an S-box using a fractional linear transformation. This method implemented the Galois field with 112 as its minimum and maximum nonlinearity value. The average SAC was 0.510254. The research work [38] developed an S-box based on 2D multiple collapse chaotic maps with a selective self-scrambling. This method had a nonlinearity value of 106 minimum and 108 maximum. An average SAC value of 0.4673. An S-box was designed by [39] using a square polynomial transformation and permutation. The minimum and maximum nonlinearity values are 110 and 112, respectively, with an average value of SAC of 0.5. Using a Möbius transformation and a bit-wise shift permutation, [7] created an S-box with an SAC average of 0.5044, and a nonlinearity of 112, respectively.

This paper proposes a lightweight stream cipher; thus, we implemented the linear fractional (Möbius) transformations for the S-box design. This linear transformation is particularly used in lightweight cryptography for low-power devices. The detail of the implemented S-box is elaborated in Section 3.

3. Proposed cipher design

This section presents the methodology for the proposed lightweight stream cipher to protect visually captured data by IoT devices. This section also evaluates the designed S-box, pseudocode, and computational complexity.

Fig. 1 shows the methodology of the proposed *SuPOR* cipher with five steps of cryptographic operations, including key generation steps. The *SuPOR* cipher takes one video frame at a time for pixel encryption and *first* performs a pixel substitution by replacing the video pixels with the designed S-box. *Secondly*, a pixel permutation is applied by shuffling the substituted video pixels. *Thirdly*, the result is then XORed with a randomly generated 64-bit one-time key length, and the output is converted to a byte array. *Fourthly*, a pixel right circular shift was used in the outcome, and *Fifth*, pixel swapping was performed on the video pixels. These steps are elaborated below.

3.1. Pixel substitution

An S-box is a component used in many cryptographic algorithms to perform a non-linear substitution of input values. It takes a fixed number of input bits and produces a corresponding fixed number of output bits, according to a predefined substitution table. In this step, a new substitution box (S-box) was designed, and the video pixels are substituted with the S-box values to allow confusion and diffusion effects in the ciphertext. The strength of a cryptosystem [40] is primarily determined by its ability to withstand attacks (differential), so a strong S-box is mandatory to construct a robust cipher.

3.1.1. The S-box design

The S-box was constructed by applying Möbius transformations and Galois fields. The Möbius transformations are represented in (1) in matrix form, consisting of translation, inversion, rotation, and dilation, as given in (2).

$$\Phi(x) \approx \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} \tag{1}$$

$$\begin{aligned} f_1(x) &= x + \frac{d}{x} \\ f_2(x) &= \frac{1}{x} \\ f_3(x) &= -\frac{ad-bc}{c^2} \cdot x \\ f_4(x) &= x + \frac{a}{c} \end{aligned} \tag{2}$$

From (2) multiply the functions to generate the Möbius transformation formula in (3).

$$f_4(x) \cdot f_3(x) \cdot f_2(x) \cdot f_1(x) = f(x) = \frac{ax+b}{cx+d} \tag{3}$$

Step 1: Considering the projective general linear group ($\mathcal{PGL}$) with coefficients F according to (4) where $n = 2$ and F = set of finite fields, can be treated as fractional linear transformations.

$$\mathcal{PGL}(n, F) \tag{4}$$

The video pixels' value range from 0 to 255; thus, let F in (4) be set of Galois field with degree 8 to give a result ranging from 0 to 255 as represented in (5) where $p = 2$ and $n = 8$

$$F = GF(p^n) \Rightarrow F = GF(2^8) \tag{5}$$

Eq. (5) has a set of 30 different irreducible polynomials to construct finite fields, and $P(n) = x^8 + x^6 + x^5 + x^4 + 1$ was randomly selected. The irreducible polynomial has square-free, monic, primitive, irreducible, and non-linear properties. However, the finite fields were calculated using (6).

$$GF(2^8) = \frac{\mathbb{Z}\{0,1\}}{P(n)} \tag{6}$$

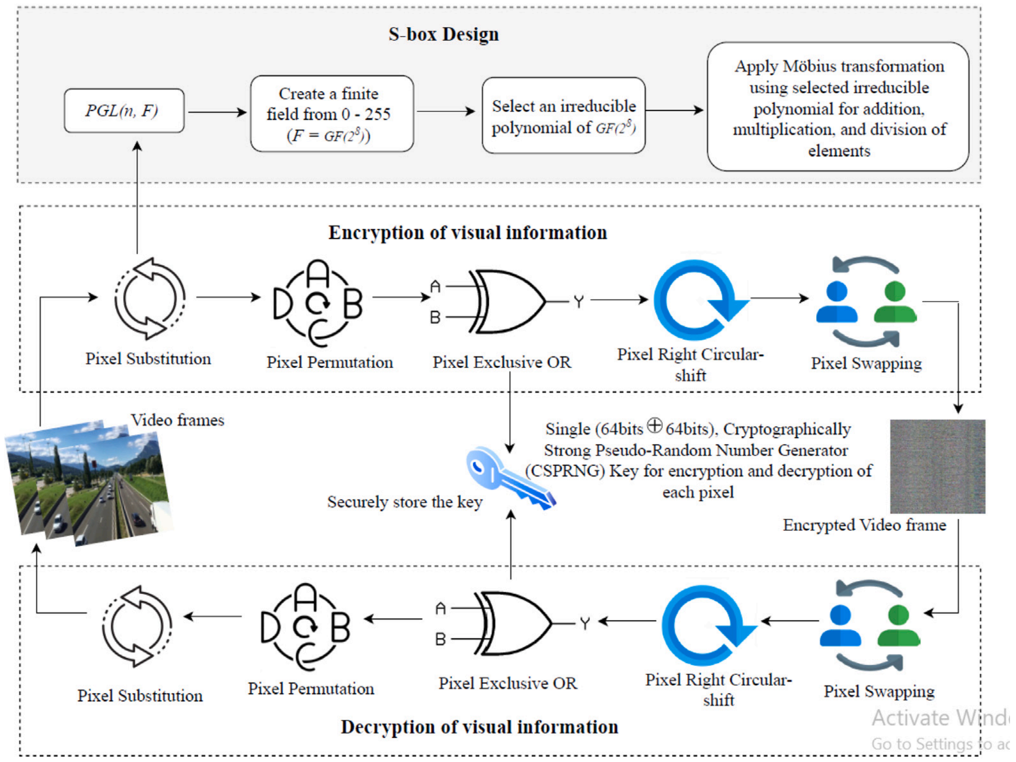

S-box Design
PGL(n, F)
Create a finite field from 0 - 255 (F = GF(2^8))
Select an irreducible polynomial of GF(2^8)
Apply Möbius transformation using selected irreducible polynomial for addition, multiplication, and division of elements
Encryption of visual information
Pixel Substitution
Pixel Permutation
Pixel Exclusive OR
Pixel Right Circular-shift
Pixel Swapping
Video frames
Single (64bits ⊕ 64bits), Cryptographically Strong Pseudo-Random Number Generator (CSPRNG) Key for encryption and decryption of each pixel
Securely store the key
Encrypted Video frame
Decryption of visual information
Activate Wind
Go to Settings to ac

Fig. 1. Methodology of the proposed *SuPOR* Cipher.

Table 1
S-box construction using Möbius transformations.

x	$f(x) = \frac{ax+b}{cx+d}$	$S-boxElements$
0	$f(0) = \frac{45(0)+25}{8(0)+4}$	90
1	$f(1) = \frac{45(1)+25}{8(1)+4}$	212
2	$f(2) = \frac{45(2)+25}{8(2)+4}$	174
253	$f(253) = \frac{45(253)+25}{8(253)+4}$	252
254	$f(254) = \frac{45(254)+25}{8(254)+4}$	171
255	$f(255) = \frac{45(255)+25}{8(255)+4}$	42

Substituting (6) into (5) and also substituting (5) into (4) gives (7)

$$PGL(2, \frac{\mathbb{Z}\{0,1\}}{P(n)}) \tag{7}$$

Step 2: Apply the Möbius transformations on (7) where $a, b, c, d \in GF(2^8)$ and $ad - bc \neq 0$. The selected values are shown in (8) where x ranges from 0 to 255, as shown in Table 1

$$f(x) = \frac{ax+b}{cx+d} \Rightarrow f(x) = \frac{45x+25}{8x+4} \tag{8}$$

As a result of applying the Möbius transformations to x from 0 to 255, the result of the S-box generated for the pixel substitution can be seen in Table 2.

3.1.2. Statistical analysis of the S-box

There are several statistical criteria that can be used to test the strength of an S-box, these are (a) nonlinearity, (b) strict avalanche criteria (SAC), and (c) bit independence criteria (BIC). The designed S-box was tested with these criteria.

Nonlinearity: This causes the output of the S-box to be uncertain by providing resistance against linear and differential attacks [37]. The nonlinearity of the S-box is calculated using the Walsh spectrum in (9). Nonlinearity is related to performance and the higher it is, the better. The nonlinearity result of the S-box is shown in Table 3.

$$NL = \frac{1}{2}(2^n - WHT(Max(f)) \tag{9}$$

Strict Avalanche Criteria (SAC): The SAC implies that if any input bit is flipped, then exactly half of the output bits should change; hence ≥ 0.5 is considered better. This is used to measure how much confusion there is between the key and the cipher-pixels. The result of the proposed S-box SAC is rounded in the third place and is analysed in Table 4 with an average of 0.5054.

Bit Independence Criteria (BIC): According to BIC, each input bit "i" should invert independently for all the output bits "j" and "k". This examines the correlation between the pixels and the cipher-pixels bits. This article performs the BIC for the nonlinearity in Table 5 and the SAC in Table 6 with values rounded off at 3rd place.

3.2. Pixel permutation

In this step, the order and pattern of the substituted pixels were scattered without repetition as stated in (10) where ∈ represents elements of natural numbers $\mathbb{N}$, $P_r(n, q)$ is the permutation function, n is the total of elements (256) and q is the number of chosen elements (256).

$$\begin{aligned} &n, q \in \mathbb{N} \rightarrow \mu_{11}, \mu_{12}, \mu_{13}, \dots, \mu_{1q}, \mu_{21}, \dots, \mu_{n-1}q, \mu_{n1}, \mu_{n2}, \mu_{n3}, \dots, \mu_{nq} \\ &f_{per} :\rightarrow P_r(n, q) = \frac{(n-q)!}{n!} \end{aligned} \tag{10}$$

3.3. Pixel XOR and key generation

This section discusses the key generation process and the question of combining the pixel with the key.

3.3.1. Key generation

In this algorithm, the encryption key was generated with two 64-bit random one-time integers produced by a cryptographically secure pseudo-random number generator (CSPRNG); these numbers are XORed with each other to produce a single key value as shown in (11).

Table 2
S-box implemented for pixel substitution.

90	212	174	70	15	39	209	87	0	127	66	173	46	231	255	189
137	115	124	183	236	248	191	198	175	226	24	155	86	223	150	220
121	216	77	144	47	208	120	172	93	135	188	35	68	82	103	234
163	74	28	49	143	184	119	25	160	247	17	88	204	105	101	186
91	116	22	193	80	166	197	246	187	130	96	97	5	149	21	237
113	131	11	13	34	219	69	57	58	239	123	207	29	72	138	106
200	26	170	118	146	228	9	217	134	33	148	202	240	40	125	7
107	43	84	16	179	222	3	182	177	6	159	111	44	55	249	89
213	73	136	181	51	41	32	12	27	141	224	19	199	110	142	151
99	20	251	30	61	133	36	1	95	158	83	227	56	92	168	129
242	117	59	169	31	165	162	132	122	98	180	229	2	67	190	140
48	221	192	201	81	178	250	241	147	79	253	8	164	53	102	65
195	4	206	238	114	232	100	63	176	50	254	161	38	210	128	78
185	157	85	62	37	225	145	214	215	139	156	71	18	108	211	194
196	64	152	75	112	233	218	23	235	244	104	153	167	60	109	203
126	205	76	10	54	94	154	52	245	230	45	14	243	252	171	42

Table 3
Nonlinearity result of the proposed S-box.

S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	*Average*
112	112	112	112	112	112	112	112	112

Table 4
SAC for the proposed S-box.

0.547	0.484	0.500	0.485	0.531	0.469	0.516	0.547
0.485	0.500	0.484	0.531	0.516	0.516	0.547	0.547
0.500	0.484	0.531	0.516	0.516	0.531	0.547	0.484
0.484	0.531	0.516	0.500	0.484	0.516	0.484	0.500
0.531	0.516	0.500	0.484	0.516	0.516	0.500	0.484
0.516	0.500	0.484	0.500	0.453	0.531	0.484	0.531
0.500	0.484	0.500	0.453	0.484	0.547	0.531	0.516
0.484	0.500	0.453	0.531	0.500	0.469	0.516	0.500

Table 5
BIC for the proposed S-box nonlinearity.

0	112	112	112	112	112	112	112
112	0	112	112	112	112	112	112
112	112	0	112	112	112	112	112
112	112	112	0	112	112	112	112
112	112	112	112	0	112	112	112
112	112	112	112	112	0	112	112
112	112	112	112	112	112	0	112
112	112	112	112	112	112	112	0

Table 6
BIC for the proposed S-box SAC.

0	0.500	0.500	0.500	0.500	0.500	0.500	0.500
0.500	0	0.495	0.495	0.509	0.509	0.463	0.463
0.500	0.495	0	0.495	0.486	0.519	0.486	0.519
0.500	0.495	0.495	0	0.522	0.474	0.474	0.522
0.500	0.509	0.486	0.522	0	0.509	0.486	0.522
0.500	0.509	0.519	0.474	0.509	0	0.474	0.519
0.500	0.463	0.486	0.474	0.486	0.474	0	0.463
0.500	0.463	0.519	0.523	0.523	0.519	0.463	0

CSPRNG offers effective cryptographic security for random numbers, and a Python module "secrets" was employed to generate the $64-\mathrm{bit}$ random key. The key is not reused throughout the encryption process; thus, this prevents power or timing side-channel attack, because an attacker cannot predict the key, given its one-time use.

$$2^{64} = random(0, 2^{64}) \bigoplus random(0, 2^{64}) \tag{11}$$

3.3.2. NIST test for key generation

The NIST (National Institute of Standards and Technology) random bit generation test is a series of statistical tests used to evaluate the random bit sequences produced by the CSPRNG generator, ensuring the security of cryptographic protocols [41]. The guidelines and full details of these tests are provided in the NIST SP 800-22 [41]. To pass, the p-value must be greater than 0.025. In this paper, four keys generated by the *SuPOR* cipher were randomly selected to assess their statistical randomness and strength using the frequency (monobit) and runs test suite from NIST SP 800-22. The results are presented in Table 7, which indicates that the CSPRNG numbers passed the tests.

3.3.3. Mask-based key storage

The proposed *SuPOR* cipher generates its key once without reusing and stores the randomly generated 64-bit key by masking the pixels in each frame for naïve or selective encryption, and stores the keys in the respective masked pixels as shown in (12).

$$\begin{aligned} &masked \rightarrow np.zeros(framesize) \\ &masked \leftarrow key \end{aligned} \tag{12}$$

The masked-key frames are saved as a video and are securely stored separately in an immutable hardware wallet or vault, which is loaded along with the encrypted videos for decryption. Another encryption that uses a one-time key is the One-time Pad (OTP) [42], a randomly generated, secure, unbreakable encryption commonly used with text data. An OTP is always the same length as the text, and its key is typically stored on a pad [43]. However, the advantage of *SuPOR* over OTP is that it is applied to visual data, with a $64-\mathrm{bit}$ key for each pixel in the video frame, and these keys are stored using masked-based key storage.

3.3.4. Pixel XOR

The video colour resolutions are mostly 8-bit, giving a colour shade of 2^8 with a total of 256. This means that the colours will fall in the range of 0 to 255.

In this step, each of the colour values generated from the output of the pixel permutation was XORed with the randomly generated 64-bit one-time key, and then the result is converted to a byte array as shown in (13).

$$\begin{aligned} &p_{XOR} = f_{per} \bigoplus key \\ &f_{XOR} : \rightarrow p_{XOR} \iff bytearray \end{aligned} \tag{13}$$

3.4. Pixel right circular-shift

Applying the right circular-shift, the last position elements of the byte array are reintroduced at the first position as the elements are shifted across an axis as shown in (14) where n is the byte array to rotate, d is the rotation space =9, and bl is the length of the byte array of 255.

$$f_{cir} : \rightarrow (n \gg d) | (n \ll (bl - d)) \tag{14}$$

Table 7
The monobit and runs NIST statistical test on the generated key.

CSPRNG number	Monobit test p-value	Status	Runs test p-value	Status
11205419218665563072	0.61708	Pass	0.97477	Pass
12472443624617409606	1.00000	Pass	1.00000	Pass
10173069113912321367	0.80259	Pass	0.79475	Pass
4649277161804844773	0.89974	Pass	0.90050	Pass

3.5. Pixel swap

This is an exchange of the positions of the byte array obtained from the previous step with each other as described in (15).

$$f_{swa} :\rightarrow A, B = B, A \tag{15}$$

3.6. Decryption of SuPOR cipher

The decryption process of *SuPOR* for retrieving the original data is the reverse of its encryption process shown in Fig. 1.

3.7. Complexity computation of SuPOR (per video frame)

The pseudo-code representation of the proposed *SuPOR* cipher is presented in the "Algorithm 1". The *SuPOR* algorithm consists of multiple steps per frame, **Step 1** loops through every pixel of the frame to replace with the S-box values, therefore, consumes $Big(O) = framesize$, for $frame\_width \times frame\_height$ video. **Step 2** performs pixel permutation which takes $Big(O) = n$, for the number of pixels n. **Step 3** loops through every pixel of the frame to XOR the pixels with the key and generate a byte array frame of the same size; therefore, the time complexity of this step is $Big(O) = framesize$, for $frame\_width \times frame\_height$ video. In **Step 4**, the algorithm performs a circular right shift on the byte array pixels which results in $Big(O) = n \times m$, for n number of pixels (byte-array) and m number of rotation places. **Step 5** loops through the frame and performs a swap on the position of the byte array which takes $Big(O) = framesize$, for $frame\_width \times frame\_height$. This sums up to $Big(O) = nm + n(3 \times framesize)$ which is approximately a linear-time complexity.

Algorithm 1: Pseudo-code of *SuPOR* stream cipher

```
Input: A video frame
Output: Video with SuPOR encryption
Data: Load Video from path
while $video == True$ do
    /* Step 1: Pixel Substitution */
    for $i$ in range $len(frame\_pixels)$ do
        $pixel \leftarrow (S - box)$
        $frame\_pixels[i] \leftarrow pixel[frame\_pixels[i]]$
    /* Step 2: Pixel Permutation */
    $pix \leftarrow array(frame\_pixels)$
    $ind \leftarrow permutation(len(pix))$
    $shuffle \leftarrow pix[ind].np.uint8$
    /* Step 3: Pixel XOR */
    $k_1 \leftarrow secrets.randbits(64)$
    $k_2 \leftarrow secrets.randbits(64)$
    $key \leftarrow k_1 \bigoplus k_2$
    for $index$ , $values$ in enumerate $shuffle$ do
        $shuffle[index] \leftarrow values \bigoplus key$
        $frame\_byte \leftarrow bytearray(shuffle)$
    /* Step 4: Pixel Right Circular-shift */
    $circular \leftarrow (frame\_byte, rotationspace = 9)$
    /* Step 5: Pixel Swap */
    for $j$ in range $(len(circular), 2)$ do
        if $(j < len(circular))$ then
            $frame\_swap[i], frame\_swap[i+1] \leftarrow circular[i+1], circular[i]$
$cv2.destoryAllWindows()$
$video.release()$
Output: Encrypted video frame with SuPOR
```

4. Evaluation

This section describes the experimental details and the characteristics of the SOTA ciphers.

A complete IoT testbed was configured for the experiments using Raspberry Pi 4 [44] and Intel NUC mini computers [45]; specifications are provided in Table 8. *SuPOR* and SOTA stream ciphers were implemented on the IoT testbed in the Python programming language. The experiments were carried out on a dataset of six (06) publicly available videos (fixed camera/static (04) and moving cameras/dynamic (02)) from the databases [46–48]. Each selected test video has varying features, that is, colours, motion activity, and spatial information. The properties of these test videos are described in Table 9.

Table 8
Experimental set up for *SuPOR*.

Device	Raspberry Pi 4	Intel NUC
Model	Model B (Rev 1.1)	NUC11TNHi7
Processor	ARMV7 rev3(v7l)	Core i7-1165G7 Quad-core
System speed	1.50 GHz	2.80 GHz × 8
Installed RAM	4.0 GB	16.0 GB
Operating system	Linux	Ubuntu 22.04.2 LTS
Graphics	VideoCore VI GPU	TGL GT2

Table 9
Properties of the datasets used for the experiments.

Video file	Size (MB)	Resolution	Time (s)	Background	Frame count
Highway	2.70	1280 × 720	5	Static	127
PET	0.99	768 × 576	12	Static	84
Ped	2.43	854 × 480	11	Static	265
Mall	2.35	960 × 540	12	Static	200
Horse_move	4.29	860 × 484	5	Dynamic	126
Safari	6.00	1280 × 720	5	Dynamic	120

4.1. Attributes of the SOTA stream ciphers

As mentioned in the introduction section, the SOTA stream ciphers, i.e., Chacha20, AES-CFB, Salsa20 and XOR were also computed for performance testing of the proposed *SuPOR* stream cipher. AES is a well-known industry standard and can be operated as both block cipher and stream cipher depending on the mode selected. For video experiments in this paper, AES with CFB mode (a stream cipher mode) was implemented. CFB mode was chosen because of its self-synchronising properties for video streams. The analysis of the SOTA stream ciphers is based on the number of operations, rounds, and cryptography principles executed as shown in Table 10.

4.1.1. Operations per cipher

The number of operations executed in each round of the ciphers is given in Table 10. The total amount of operations performed on a single video frame is given in (16). Here, (OE) is the execution of the operations and $\mathcal{T}(OE)$ is the total number of operations executed.

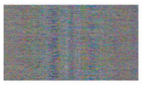
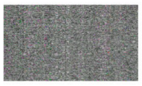
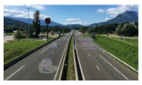
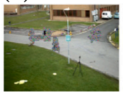
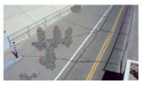
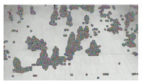
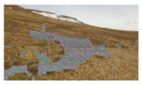
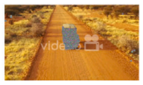
$$\mathcal{T}(OE) = OE\_in\_a\_Round \times Rounds \tag{16}$$

It can be easily confirmed from (16) that the total number of operations in each frame depends on the number of rounds. Ciphers with more rounds need to execute more operations (with a huge number

Table 10
Attributes of SOTA stream ciphers.

Ciphers	Operations in each round	Rounds	Cryptography principles
AES-CFB	4 (5 in last round)	10	SubBytes, ShiftRows, MixColumns, Add round key
Chacha20	3	20	Addition module, Bitwise exclusive-or (XOR), N-bit left rotation
Salsa20	3	20	32-bit addition, 32-bit XOR, Constant-distance of 32-bit rotation
XOR	1	1	Bitwise exclusive-or (XOR)
Proposed ***SuPOR***	5	1	Substitution, Permutation, Bitwise exclusive-or (XOR), Right circular shift, Swap

Table 11
Visual representation of the original (O), naïve encrypted (NE), and selectively encrypted (SE) frames using *SuPOR* cipher.

Highway	PET	Ped	Mall	Horse_move	Safari
a(1) $O - Frame : 85$	**a(2)** $O - Frame : 30$	**a(3)** $O - Frame : 90$	**a(4)** $O - Frame : 60$	**a(5)** $O - Frame : 45$	**a(6)** $O - Frame : 100$
 **b(1)** $NE\ Video$	**b(2)** $NE\ Video$	**b(3)** $NE\ Video$	 **b(4)** $NE\ Video$	**b(5)** $NE\ Video$	**b(6)** $NE\ Video$
 **c(1)** $SE\ Video$	 **c(2)** $SE\ Video$	 **c(3)** $SE\ Video$	 **c(4)** $SE\ Video$	 **c(5)** $SE\ Video$	 **c(6)** $SE\ Video$

of instructions) per round, making them computationally expensive in comparison with proposed *SuPOR* which is a single-round cipher.

5. Results and discussion

This section examines the visual result and resistance of *SuPOR* to various attacks with reasonable computational cost, and the comparative evaluation of SOTA ciphers with *SuPOR*.

5.1. Visual results of SuPOR cipher

The visual results of the implementation of our *SuPOR* cipher in the surveillance videos are given in Table 11. According to Table 11, each video frame is selected from the videos tested derived from static and dynamic IoT camera devices by showing the original frame in row 1 (a(1 - 6)). The results show the naïve encryption and selective (foreground (FG)) encryption (SE) of the frames using *SuPOR* cipher in row 2 (b(1 - 6)) and row 3 (c(1 - 6)), respectively. SE is applied to moving objects (FG) in videos that were extracted (before encryption) using the Gaussian mixture method (GMM) [17] for static background videos and Advanced Flow of Motion Detection (AFOM) [19] for dynamic background videos.

5.2. Security analysis of SuPOR cipher

This sub-section discusses the security paradigm of *SuPOR* based on the different following attacks:

5.2.1. Padding oracle attacks

This attack uses the padding validation of an encrypted message to decrypt it using a “padding oracle” who responds to queries about whether a message is correctly padded or not. *SuPOR* performs encryption only on the available number of pixels in the frame without padding, thus preventing padding oracle attacks.

5.2.2. Slide attacks

The slide attack works by analysing the key schedule and exploiting its weaknesses to break the cipher. This is common with encryption that applies key repetition in a cyclic manner [49].

SuPOR cipher implemented a randomly generated key once and does not repeat these keys, making it secure against slide attacks.

5.2.3. Key sensitivity attacks

The purpose of this technique is to test whether a slight modification of the secret key can decrypt the video frame. This test was applied to two different video frames (highway frame 110 and horse_move frame 95) as shown in Fig. 2. Fig. 2 (a1) and (b1) with a modified encryption key for decryption, did not reproduce or correlate with the original frame as shown in Fig. 2(a) and (b), therefore, a change in the key cannot decrypt the video frames. This implies that the*SuPOR* cipher can withstand the key modification attack.

5.2.4. Chosen plaintext attacks

In this attack, the attacker can determine the ciphertext for any arbitrary plaintext by analysing the encryption of the plaintext to find the corresponding ciphertext, thus compromising the encryption scheme’s security. *SuPOR* was tested against this attack by choosing a plain video (white pixels) and thereby encrypting it and checking if an attacker can predict what the encrypted video pixels are from the plain video pixels, as shown in Fig. 3. Fig. 3(b) confirms that an attacker cannot predict or analyse information about pixels since there is no interconnection with Fig. 3(a).

5.2.5. Key guessing or brute-force attacks

This involves cracking the encryption key using trial and error methods until the correct key is revealed. The key implementation in *SuPOR* cipher randomly generates two different 64-bit one-time integers with CSPRNG which are then XORed together and applied to each of the pixels given $2^{64+total\_pixel\_number}$. The horse_move video has

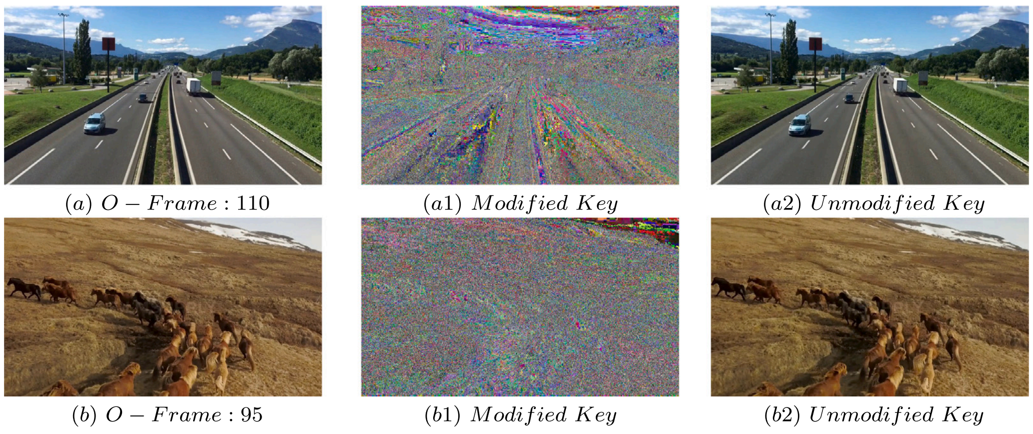


Fig. 2. Key sensitivity attack on test videos (a, b) Original (O) video frames, (a1, b1) Decrypted video frame with key modification, (a2, b2) Decrypted video frame without key modification.

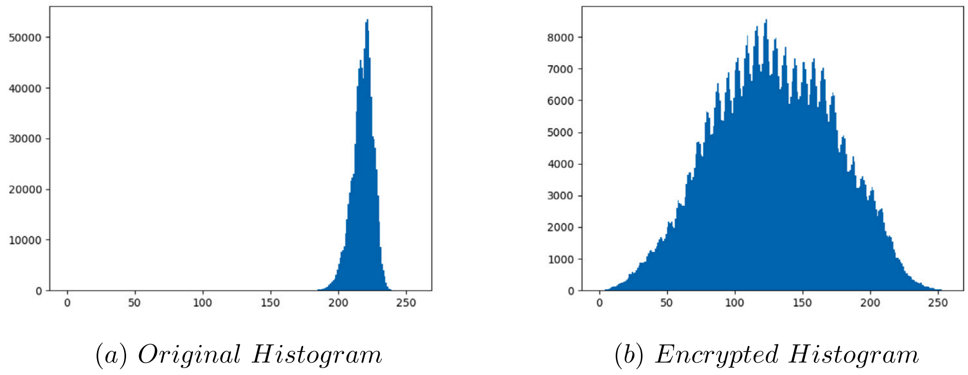


Fig. 3. Visual representation of chosen plain attack on plain video (with all white colour pixels) (a) Histogram analysis of the original video frame # 85 (b) Histogram analysis of the encrypted video frame # 85.

416 240 pixels (860 × 484), therefore the total key will be 2^{416304} as shown in (17), making it resistant to brute-force attack.

$$\begin{aligned} &2^{64} = random(2^{64} \bigoplus 2^{64}) \\ &Key :\rightarrow 2^{64} \times 2^{416240} \\ &Key :\rightarrow 2^{64+416240} \\ &Key :\rightarrow 2^{416304} \end{aligned} \tag{17}$$

5.2.6. Differential attacks

In this attack, the corresponding key or pixels (plain-pixels) are determined by comparing the variations in the pixel input and the encrypted pixel output. This attack was implemented by changing a single bit (pixel) of the original video frame to evaluate its effect when encrypted using the Number of Pixel Changing Rate (NPCR) in (18) and Unified Average Changing Intensity (UACI) in (19) where T is total pixels in the video frame, HPV denotes highest pixel value (i.e., 255), $\mathcal{V}f^e$ is the unaltered video frame pixel encryption and $\mathcal{V}f^{e*}$ is the altered video frame pixel encryption. The NPCR and UACI values should be ≥99 and ≥33 respectively, to have a high level of resistance to differential attacks. For the experiment, a single video frame was selected, and one of the pixel values was changed, encrypted, and compared with the same frame without pixel change using the NPCR and UACI equations. The result is displayed in Table 12 with the frame number that was altered.

$$\begin{aligned} &\mathrm{NPCR}(\mathcal{V}f^e, \mathcal{V}f^{e*}) = \left[\frac{\sum_{1\le j\le n}^{1\le i\le m} \mathbb{P}(i,j)}{T} \right] \times 100, \\ &\mathbb{P}(i,j) = \begin{cases} 0, & \text{if } \mathcal{V}f^e(i,j) = \mathcal{V}f^{e*}(i,j) \\ 1, & \text{if } \mathcal{V}f^e(i,j) \neq \mathcal{V}f^{e*}(i,j) \end{cases} \end{aligned} \tag{18}$$

$$\mathrm{UACI}(\mathcal{V}f^e, \mathcal{V}f^{e*}) = \frac{1}{T}\left[\frac{\sum_{1\le j\le n}^{1\le i\le m} \ |\mathcal{V}f^e(i,j) - \mathcal{V}f^{e*}(i,j)|}{HPV} \right] \times 100 \tag{19}$$

Based on the results of the differential attack in Table 12, the proposed *SuPOR* cipher shows that it is significantly more resistant to the differential attack compared to the SOTA stream ciphers discussed (Table 10). Bentahar [50] also confirms the low values of NPCR and UACI of these ciphers, making them inadequate for visual protection in IoT devices.

5.3. Statistical analysis of *SuPOR* cipher

This sub-section describes the statistical analysis of videos after applying *SuPOR*. Statistical analysis is beneficial for identifying trends and developing valuable insights on the pixel distribution and correlation in the original and relevant decrypted pixels. The statistical analysis of videos after applying *SuPOR* is given below:

5.3.1. Histogram analysis

Histogram analysis is a graphical representation of the diffusion in the pixel intensity of a video frame in a greyscale. In greyscale, each pixel's value is a single sample, which means it only contains intensity data for pixels with values ranging from 0 to 255. An encrypted video frame with a good cipher produces a uniformly-distributed histogram analysis. In this article, histogram analysis was applied to the original and encrypted video frame with the result displayed in Table 13. Table 13 a(1–6) does not mirror Table 13 b(1–6), hence *SuPOR* cipher achieved the goal and objective of encryption.

Table 12
Results on Differential attack performed on the video frame.

Video		NPCR					UACI				
Files	Frame	AES-CFB	Chacha 20	Salsa 20	XOR	***SuPOR***	AES-CFB	Chacha 20	Salsa 20	XOR	***SuPOR***
Highway	5	98.132	98.103	99.411	99.432	99.657	6.464	6.468	21.215	21.226	32.393
PET	0	98.214	98.247	99.433	99.436	99.660	6.904	6.897	20.988	20.917	31.746
Ped	5	98.093	98.088	99.426	99.434	99.654	6.468	6.479	21.234	21.238	32.092
Mall	5	98.094	98.133	99.430	99.409	99.768	6.474	6.466	21.238	21.232	36.300
Horse_move	5	98.073	98.103	99.435	99.430	99.709	6.462	6.468	21.215	21.094	34.639
Safari	5	98.113	98.118	99.429	99.429	99.673	6.466	6.473	21.225	21.031	32.274

Table 13
Histogram analysis of the original (O) frames and the naïve encrypted (NE) frames.

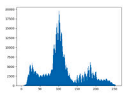
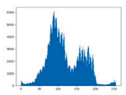
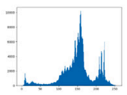
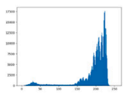
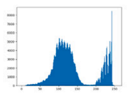
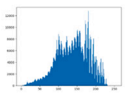
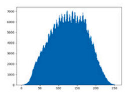
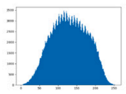
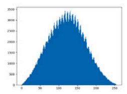
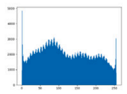
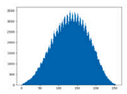
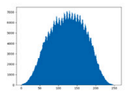
Highway	PET	Ped	Mall	Horse_move	Safari
					
a(1) $(Frame\ 5)\ O$	**a(2)** $(Frame\ 0)\ O$	**a(3)** $(Frame\ 115)\ O$	**a(4)** $(Frame\ 15)\ O$	**a(5)** $(Frame\ 20)\ O$	**a(6)** $(Frame\ 5)\ O$
					
b(1) $(Frame\ 5)\ NE$	**b(2)** $(Frame\ 0)\ NE$	**b(3)** $(Frame\ 115)\ NE$	**b(4)** $(Frame\ 15)\ NE$	**b(5)** $(Frame\ 20)\ NE$	**b(6)** $(Frame\ 5)\ NE$

Table 14
Entropy results for the *SuPOR* and SOTA stream ciphers.

Videos		AES (CFB)	Chacha 20	Salsa 20	XOR	***SuPOR***
File	Frame					
Highway	5	5.869	5.870	7.556	7.556	7.575
PET	0	5.967	5.969	7.540	7.540	7.575
Ped	115	5.810	5.785	7.472	7.553	7.602
Mall	15	5.674	5.673	7.474	7.555	7.961
Horse_move	20	5.776	5.673	7.473	7.421	7.577
Safari	5	5.869	5.871	7.556	7.542	7.554

5.3.2. Entropy analysis

In entropy analysis, a frame's entropy refers to how much uncertainty or randomness is contained in it to estimate its information content. As entropy is calculated using greyscale, a value nearer to 8 (for an 8-bit video frame) represents randomness at its maximum.

$$\mathcal{E}(Y) = -\sum_{i=0}^{x-1} p(i)\log_2 p(i) \tag{20}$$

Shannon entropy was applied to the encrypted video frame using (20) where x is the number of grey levels, $p(i)$ is the probability that a pixel has a grey level i, and the result was compared with other SOTA ciphers in Table 14. The results in Table 14 indicate that *SuPOR* has a higher entropy, making it a better choice for IoT devices.

5.3.3. Mean squared and absolute error analysis

The Mean Squared Error (MSE) measures are based on the average of the squares of the errors, which means how close the original frame is to the encrypted frame. The Mean Absolute Error (MAE) measures the average magnitude using the absolute differences between the encrypted frame and the original frame. Both MSE and MAE determine the difference between all the pixels in a frame. Generally, when the MSE and MAE values of a frame are higher, there is a greater difference between the original and encrypted frames, which means that the security of the proposed cryptosystem is stronger. The MSE was calculated with (21) while the MAE was calculated as shown in (22). where n is the number of pixels, Y_i is the original video frame, and $\hat{Y}_i$ is the encrypted video frame.

$$\mathcal{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{21}$$

$$\mathcal{MAE} = \frac{1}{n}\sum_{i=1}^{n}|(Y_i - \hat{Y}_i)| \tag{22}$$

As can be seen in Fig. 4, there is a large difference (>100) between the original and encrypted video frame, indicating the effectiveness and efficiency of the proposed cryptography.

5.3.4. Correlation analysis and video metrics

Correlation measures the extent to which variables are mutually or linearly related, and is denoted by r as described in (23). $r = +1$ indicates perfect linearity, $r = 0$ indicates "no correlation", meaning variables are independent, and $r = -1$ indicates "inverse correlation", meaning that as one variable increases the other decreases. Correlation analysis can be applied in image processing in two ways: first, on frames (original and encrypted video frames) and second, on pixels that are adjacent within the frames. The adjacent pixels of the original frame are always highly correlated, whereas this correlation diminishes or moves toward zero when using a competent encryption algorithm.

$$r = \frac{\sum(x_i - \hat{x}_i)(y_i - \hat{y}_i)}{\sqrt{\sum(x_i - \hat{x}_i)^2\sum(y_i - \hat{y}_i)^2}} \tag{23}$$

The video metric is measured in this paper using the Structural Similarity Index (SSIM) which compares two images with similar structures to determine the perceptual difference between them. The SSIM value ranges from −1 to 1, and if two images are almost identical, their SSIM will be close to 1. The SSIM was compared with the decrypted frames and the original frames

The pixel correlation was applied on randomly selected 200 column-pixels in the original and encrypted frame separately, and the whole frame correlation of both frames, as documented in Table 15 as well as the SSIM. In the frame correlation column in Table 15 $r = 0$ which shows no correlation between the encrypted and original frame, in the pixel correlation column, the encrypted values are negative, this proves that *SuPOR* cipher has high security against statistical attack. In addition, the value on the SSIM column is close to 1 which means that the decrypted frames are lossless.

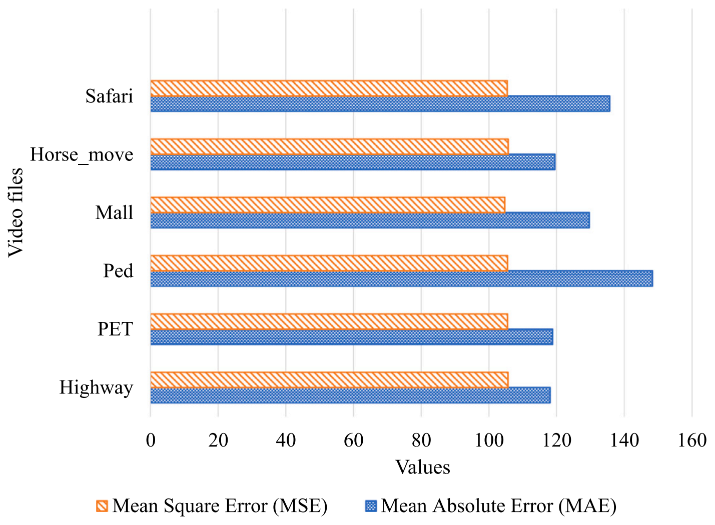


Fig. 4. The Analysis of MSE and MAE on original and encrypted frames.

Table 15
Frame and pixel correlation, with SSIM analysis on frames.

Videos		Frame	Pixel correlation		SSIM
File	Frame	Correlation	Original	Encrypted	
Highway	100	−0.00409	0.66609	−0.00691	0.94837
PET	65	0.01804	0.63881	−0.00956	0.9284
Ped	85	0.07402	0.37217	−0.00829	0.90271
Mall	115	−0.00232	0.06058	−0.04382	0.94034
Horse_move	20	−0.03723	0.51324	−0.07600	0.95859
Safari	80	0.00438	0.43423	−0.00902	0.95667

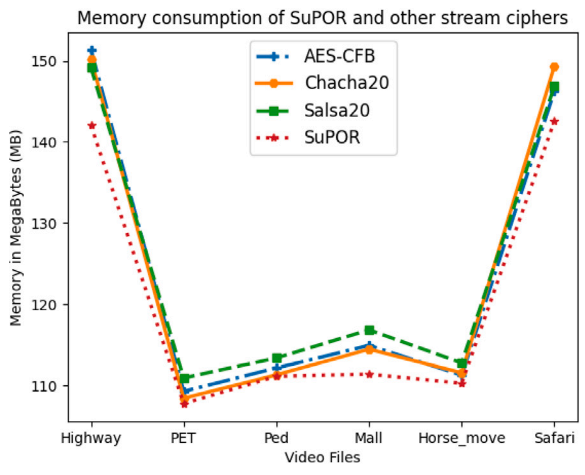


Fig. 5. Memory consumption analysis of *SuPOR* and other SOTA ciphers.

5.4. Comparative evaluation with SOTA stream ciphers

This sub-section comparatively evaluates the performance of *SuPOR* with SOTA stream ciphers (AES-CFB, Chacha20, Salsa20, XOR) on the basis of total operations per round, memory consumption, and computational analysis. This comparative evaluation was computed for naïve encryption only on tested videos.

5.4.1. Memory consumption

Memory consumption is a measure of the space allocated for the implementation of the algorithms, which is determined by the total number of instructions executed in each algorithm, the generation and size of keys and the mode of operations in these algorithms.

The memory consumption of the proposed *SuPOR* was evaluated and compared with the SOTA stream ciphers i.e., AES-CFB, Chacha20, and Salsa20 in Fig. 5. Fig. 5 shows the memory consumption of computed ciphers based on five average observations taken in megabytes (MB). *SuPOR* has the lowest memory consumption compared to the other ciphers due to the smaller number of instructions executed in its single encryption round. The reason for doing this test with naïve encryption is because stream ciphers usually perform slower to encrypt a large lump of visual data. Fig. 5 demonstrated that *SuPOR* is reasonably efficient for naïve encryption on constrained devices.

5.4.2. Computational analysis

Based on the total number of operations and memory consumption discussed in (16) and Fig. 5, respectively, the analysis of processing time and energy consumption was performed considering encryption and decryption timings. The encryption time indicates how long it takes for the algorithm to convert the plain-pixels into the cipher-pixels based on the key size and mode of operation, while the decryption time is the time it takes to return the cipher-pixels to plain-pixels.

The results were taken using encryption and decryption timings in microseconds (μs) in all the videos tested. Table 16 shows the comparative results for Raspberry Pi and Table 17 for the Intel NUC findings for all computed ciphers. On average, five observations were taken for the timings. The Raspberry Pi encryption timings in Table 16, show that *SuPOR* is faster than AES-CFB and with a close margin with Chacha20. In addition, the Intel NUC encryption timings in Table 17, the *SuPOR* cipher is faster than AES-CFB, Chacha20, Salsa20 and with a close margin with XOR. The results prove the lightweight nature of the proposed *SuPOR* stream cipher to secure visual IoT data without draining devices.

5.5. Comparative analysis of *SuPOR* with existing approaches

The *SuPOR* cipher was also compared with other existing approaches, as illustrated in Table 18 with X, indicating that the authors did not mention or specify. The NPCR and UACI results of the study [10] were based on one image for its testbed. In addition, the highest NPCR and UACI results of the study [6] are selected. However, *SuPOR* was implemented in six test videos and had high NPCR and UACI for the *Mall* video compared to [6]. Additionally, *SuPOR* showed promising results in IoT devices with only one round of encryption, unlike other approaches.

Table 16
Computational analysis (Encryption and decryption timing) of *SuPOR* and other SOTA ciphers on Raspberry Pi.

Videos	Encryption (μs)					Decryption (μs)				
	AES-CFB	Chacha 20	Salsa 20	XOR	***SuPOR***	AES-CFB	Chacha 20	Salsa 20	XOR	***SuPOR***
Highway	942 060	77 610	60 510	35 390	100 370	945 660	72 950	52 920	41 990	99 510
PET	452 370	37 400	31 920	17 120	45 160	449 520	36 280	25 570	17 080	44 010
Ped	418 920	34 530	27 020	16 060	33 500	430 580	36 820	23 670	15 990	32 570
Mall	526 960	45 440	34 310	20 070	46 970	534 490	41 790	29 830	19 980	46 430
Horse_move	423 100	35 590	27 600	16 170	33 720	416 790	37 280	24 050	16 050	33 320
Safari	938 870	78 040	66 270	36 200	99 160	931 630	73 240	57 140	35 940	98 640

Table 17
Computational analysis (Encryption and decryption timing) of *SuPOR* and other SOTA ciphers on Intel NUC.

Videos	Encryption (μs)					Decryption (μs)				
	AES-CFB	Chacha 20	Salsa 20	XOR	***SuPOR***	AES-CFB	Chacha 20	Salsa 20	XOR	***SuPOR***
Highway	53 950	16 230	14 550	3130	14 350	50 640	8990	5900	3090	14 980
PET	28 690	11 890	8850	1460	8990	24 060	4400	6100	1370	8770
Ped	28 460	11 350	9000	1700	7700	22 510	4140	5600	1260	7620
Mall	34 730	13 620	10 550	1920	9100	28 430	5190	5800	1620	8980
Horse_move	27 710	11 520	9010	1340	7780	22 260	4140	5800	1310	7680
Safari	57 360	15 690	13 790	3190	14 370	52 990	8950	5980	3140	14 460

Table 18
Comparative analysis of *SuPOR* cipher with the existing approaches.

Ref	Implementation model	Security methods	DT	Dataset	E	CC	NPCR	UACI	RE
(2021) [51]	Lightweight-Channel-Independent OFDM-Based Encryption Method for VLC-IoT Networks	Substitution, phase encryption for frequency-domain, and permutation for time-domain	$Intel\ Corei7$ 2 – GHz CPU	$OFDM\ signals$	NE	Low	X	X	1
(2022) [52]	Extended Type-1 Generalised Feistel Networks: Lightweight Block Cipher for IoT	Matrix representation method	$ARM\ Cortex-M3$ $STM32F103$ $Microcontroller$	$Texts$	NE	Low	X	X	25
(2022) [10]	Lightweight, Privacy-Preserving Cooperative Object Classification for Connected Autonomous Vehicles	Chaotic mapping and additive secret sharing technique	$Personal\ computer$	$Images$	NE	$High$	99.62	33.49	2
(2023) [9]	A Reversible Framework for Efficient and Secure Visual Privacy Protection	Region division, room vacating, image encryption (XOR), pixel adjustment, and pixel permutation	X	$Thumbnail\ images$	NE	Low	X	X	X
(2023) [53]	Pixel-split image encryption scheme based on 2D Salomon map	Two-dimensional Salomon map and pixel split	X	$Images$	NE	X	99.6	33.47	3
(2024) [6]	Temporal action segmentation for video encryption	ABNEA encryption algorithm and two-dimensional Gramacy&Lee map for pseudo-random sequence generation	$I5-4210\ CPU$	$Videos$	NE	X	99.62	33.49	X
Proposed *SuPOR*	**Lightweight Cipher for Visual Data Security on constrained IoT Devices**	**Substitution-Permutation-XOR-Shift-Swap**	$Raspberry\ pi,\ Intel\ NUC$	$Videos$	**NE, SE**	Low	99.77	36.30	1

Device testbed = DT, Encryption = E, Computational cost = CC, Rounds of Encryption = RE

5.6. Limitation and future work

The focus of the *SuPOR* cipher is to efficiently secure the data on IoT devices; hence, the experiments are limited to IoT device computation only. During the evaluation, data transmission scenarios are not considered; therefore, the communication overhead was not calculated and can be addressed in future research.

In the future extension of this work, *SuPOR* can be analysed under differential and linear cryptanalysis frameworks, benchmarking multi-round variants (e.g., 2-round *SuPOR*) across diverse resource constraints, and exploring parameterised configurations that allow users to select the number of rounds (customised model) based on their specific threat models and device capabilities. In addition, the security evaluation of *SuPOR* could be extended beyond traditional cryptographic attacks by focusing on its resilience against emerging machine learning-based threats. Specifically, it can be assessed how well *SuPOR* protects visual data from deep learning models that attempt to reconstruct or infer sensitive content from encrypted streams. To enhance resistance against such inference attacks [54], the integration of adversarial perturbation techniques and randomness amplification within the cipher structure can be considered. Furthermore, we intend to develop adaptive configurations of *SuPOR* tailored to the computational and security requirements of various IoT devices [55] for an optimal balance between efficiency and robust visual data protection.

6. Conclusion

This paper has proposed a relatively efficient and robust cipher (*SuPOR*), which could effectively safeguard visual data in constrained IoT devices. *SuPOR*- a single-round, lightweight stream cipher—was based on five steps of operations i.e. **Su**bstitution-**P**ermutation-X**OR**-Cirular_shift(right)-Swap with linear time complexity. As a primary contribution to the security robustness of *SuPOR*, a nonlinear S-box was designed using linear (Möbius) transformations for the pixel substitution. The visual results demonstrated the versatility and efficacy of *SuPOR* for both naïve or selective encryption videos taken from fixed and moving cameras. *SuPOR* possesses all mandatory security properties, which were validated through several analyses. It was tested against different attacks (key modification attack, differential attack, brute-force attack, slide attack) and was also subjected to statistical analysis (entropy, histogram, correlation) to verify the algorithm's effectiveness. The evaluation proved the resistance of the *SuPOR* against attacks and its efficiency as a secure cipher. Furthermore, the configuration of the real-time IoT testbed with Raspberry Pi and Intel NUC confirmed that the *SuPOR* is computationally efficient for visual encryption in IoT devices.

CRediT authorship contribution statement

Ifeoluwapo Aribilola: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Saeed Hamood Alsamhi:** Writing – review & editing, Validation, Project administration, Investigation, Formal analysis. **John G. Breslin:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition. **Mamoona Naveed Asghar:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao, A survey on security and privacy issues in internet-of-things, IEEE Internet Things J. 4 (5) (2017) 1250–1258, http://dx.doi.org/10.1109/JIOT.2017.2694844.
[2] H. Bi, J. Liu, N. Kato, Deep learning-based privacy preservation and data analytics for IoT enabled healthcare, IEEE Trans. Ind. Inform. 18 (7) (2022) 4798–4807, http://dx.doi.org/10.1109/TII.2021.3117285.
[3] A. Alwarafy, K.A. Al-Thelaya, M. Abdallah, J. Schneider, M. Hamdi, A survey on security and privacy issues in edge-computing-assisted internet of things, IEEE Internet Things J. 8 (6) (2021) 4004–4022, http://dx.doi.org/10.1109/JIOT.2020.3015432.
[4] J.-P. Yaacoub, H. Noura, O. Salman, A. Chehab, Security analysis of drones systems: Attacks, limitations, and recommendations, Internet Things 11 (2020) 100218, http://dx.doi.org/10.1016/j.iot.2020.100218, URL https://www.sciencedirect.com/science/article/pii/S2542660519302112.
[5] M.N. Asghar, N. Kanwal, B. Lee, M. Fleury, M. Herbst, Y. Qiao, Visual surveillance within the EU General Data Protection Regulation: A technology perspective, IEEE Access 7 (2019) 111709–111726, http://dx.doi.org/10.1109/ACCESS.2019.2934226.
[6] S. Gao, H.H.-C. Iu, J. Mou, U. Erkan, J. Liu, R. Wu, X. Tang, Temporal action segmentation for video encryption, Chaos Solitons Fractals 183 (2024) 114958, http://dx.doi.org/10.1016/j.chaos.2024.114958, URL https://www.sciencedirect.com/science/article/pii/S0960077924005101.
[7] I. Aribilola, B. Lee, M. Naveed Asghar, Möbius transformation and permutation based S-box to enhance IoT multimedia security, IEEE Access 12 (2024) 140792–140808, http://dx.doi.org/10.1109/ACCESS.2024.3466930.
[8] N. Mäurer, T. Guggemos, T. Ewert, T. Gräupl, C. Schmitt, S. Grundner-Culemann, Security in digital aeronautical communications a comprehensive gap analysis, Int. J. Crit. Infrastruct. Prot. 38 (2022) 100549, http://dx.doi.org/10.1016/j.ijcip.2022.100549, URL https://www.sciencedirect.com/science/article/pii/S187454822200035X.
[9] Y. Zhang, X. Ye, X. Xiao, T. Xiang, H. Li, X. Cao, A reversible framework for efficient and secure visual privacy protection, IEEE Trans. Inf. Forensics Secur. 18 (2023) 3334–3349, http://dx.doi.org/10.1109/TIFS.2023.3280341.
[10] J. Xiong, R. Bi, Y. Tian, X. Liu, D. Wu, Toward lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles, IEEE Internet Things J. 9 (4) (2022) 2787–2801, http://dx.doi.org/10.1109/JIOT.2021.3093573.
[11] I. Aribilola, M.N. Asghar, N. Kanwal, M. Fleury, B. Lee, SecureCam: Selective detection and encryption enabled application for dynamic camera surveillance videos, IEEE Trans. Consum. Electron. 69 (2) (2023) 156–169, http://dx.doi.org/10.1109/TCE.2022.3228679.
[12] J. Henriques, F. Caldeira, T. Cruz, P. Simões, A forensics and compliance auditing framework for critical infrastructure protection, Int. J. Crit. Infrastruct. Prot. 42 (2023) 100613, http://dx.doi.org/10.1016/j.ijcip.2023.100613, URL https://www.sciencedirect.com/science/article/pii/S1874548223000264.
[13] J. Daemen, V. Rijmen, The block cipher rijndael, in: J.-J. Quisquater, B. Schneier (Eds.), Smart Card Research and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 277–284.
[14] M.N. Asghar, M. Ghanbari, An efficient security system for CABAC bin-strings of H.264/SVC, IEEE Trans. Circuits Syst. Video Technol. 23 (3) (2013) 425–437, http://dx.doi.org/10.1109/TCSVT.2012.2204941.
[15] T. Obaida, A. Salim Jamil, N. Hassan, A review: Video encryption techniques, advantages and disadvantages, Webology 19 (2022) 2022–7209.
[16] R. Malladar, S. Kunte, Selective video encryption using Sattolo's encryption technique, in: 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques, ICEECCOT, 2016, pp. 268–273, http://dx.doi.org/10.1109/ICEECCOT.2016.7955228.
[17] L. Gupta, T. Sortrakul, A Gaussian-mixture-based image segmentation algorithm, Pattern Recognit. 31 (3) (1998) 315–325, http://dx.doi.org/10.1016/S0031-3203(97)00045-9, URL https://www.sciencedirect.com/science/article/pii/S0031320397000459.
[18] S. Shah, X. Xuezhi, Traditional and modern strategies for optical flow: An investigation, SN Appl. Sci. 3 (2021) http://dx.doi.org/10.1007/s42452-021-04227-x.
[19] I. Aribilola, M. Naveed Asghar, N. Kanwal, M. Samar Ansari, B. Lee, AFOM: Advanced flow of motion detection algorithm for dynamic camera videos, in: 2022 33rd Irish Signals and Systems Conference, ISSC, 2022, pp. 1–6, http://dx.doi.org/10.1109/ISSC55427.2022.9826141.
[20] H. Deng, Z. Qin, Q. Wu, Z. Guan, R.H. Deng, Y. Wang, Y. Zhou, Identity-based encryption transformation for flexible sharing of encrypted data in public cloud, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3168–3180, http://dx.doi.org/10.1109/TIFS.2020.2985532.
[21] P. Santoso, E. Rilvani, A. Trisnawan, K. Adiyarta, D. Napitupulu, T. Sutabri, R. Rahim, Systematic literature review: Comparison study of symmetric key and asymmetric key algorithm, IOP Conf. Ser.: Mater. Sci. Eng. 420 (2018) 012111, http://dx.doi.org/10.1088/1757-899X/420/1/012111.
[22] M.A. Saleh, N.M. Tahir, E. Hisham, H. Hashim, An analysis and comparison for popular video encryption algorithms, in: 2015 IEEE Symposium on Computer Applications & Industrial Electronics, ISCAIE, 2015, pp. 90–94, http://dx.doi.org/10.1109/ISCAIE.2015.7298334.
[23] V.S. Shetty, R. Anusha, D. Kumar M.J., P. Hegde N., A survey on performance analysis of block cipher algorithms, in: 2020 International Conference on Inventive Computation Technologies, ICICT, 2020, pp. 167–174, http://dx.doi.org/10.1109/ICICT48043.2020.9112491.
[24] S. Singh, P. Sharma, S. Moon, J. Park, Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions, J. Ambient. Intell. Humaniz. Comput. 15 (2017) 1–18, http://dx.doi.org/10.1007/s12652-017-0494-4.
[25] P. Panahi, C. Bayılmış, U. Çavuşoğlu, S. Kaçar, Performance evaluation of lightweight encryption algorithms for IoT-based applications, Arab. J. Sci. Eng. 46 (4) (2021) 4015–4037.
[26] Y. Nir, A. Langley, Chacha20 and poly1305 for IETF protocols, Rfc-Ed. Org. (2015) 1–45, http://dx.doi.org/10.17487/rfc7539, URL https://www.rfc-editor.org/rfc/pdfrfc/rfc7539.txt.pdf.
[27] D.J. Bernstein, The Salsa20 family of stream ciphers, New Stream Cipher Designs: The ESTREAM Finalists, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 84–97, URL https://doi.org/10.1007/978-3-540-68351-3_8.
[28] G. Paul, S. Maitra, RC4 Stream Cipher and its Variants, CRC Press, 2012, p. 311.
[29] B. Schneier, Description of a new variable-length key, 64-bit block cipher (blowfish), in: R. Anderson (Ed.), Fast Software Encryption, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994, pp. 191–204.
[30] L.R. Knudsen, G. Leander, PRESENT– block cipher, in: Encyclopedia of Cryptography and Security, Springer US, Boston, MA, 2011, pp. 953–955, http://dx.doi.org/10.1007/978-1-4419-5906-5_605.
[31] E. Barker, A. Roginsky, Transitioning the use of cryptographic algorithms and key lengths, 2019, http://dx.doi.org/10.6028/NIST.SP.800-131Ar2.
[32] S. Kansal, M. Mittal, Performance evaluation of various symmetric encryption algorithms, in: 2014 International Conference on Parallel, Distributed and Grid Computing, 2014, pp. 105–109, http://dx.doi.org/10.1109/PDGC.2014.7030724.

[33] A.H. Zahid, L. Tawalbeh, M. Ahmad, A. Alkhayyat, M.T. Hassan, A. Manzoor, A.K. Farhan, Efficient dynamic S-box generation using linear trigonometric transformation for security applications, IEEE Access 9 (2021) 98460–98475, http://dx.doi.org/10.1109/ACCESS.2021.3095618.

[34] A.K. Farhan, R.S. Ali, H. Natiq, N.M.G. Al-Saidi, A new S-box generation algorithm based on multistability behavior of a plasma perturbation model, IEEE Access 7 (2019) 124914–124924, http://dx.doi.org/10.1109/ACCESS.2019.2938513.

[35] Lu, Zhu, Wang, A novel S-box design algorithm based on a new compound chaotic system, Entropy 21 (10) (2019) 1004, http://dx.doi.org/10.3390/e21101004.

[36] Ü. Çavuşoğlu, A.H. Kökçam, A new approach to design S-box generation algorithm based on genetic algorithm, Int. J. Bio-Inspired Comput. 17 (1) (2021) 52–62, http://dx.doi.org/10.1504/IJBIC.2021.113360, arXiv:https://www.inderscienceonline.com/doi/pdf/10.1504/IJBIC.2021.113360. URL https://www.inderscienceonline.com/doi/abs/10.1504/IJBIC.2021.113360.

[37] S. Farwa, T. Shah, L. Idrees, A highly nonlinear S-box based on a fractional linear transformation, SpringerPlus 5 (2016) 1–12, http://dx.doi.org/10.1186/s40064-016-3298-7.

[38] C. Yang, X. Wei, C. Wang, S-box design based on 2D multiple collapse chaotic map and their application in image encryption, Entropy 23 (2021) 1312, http://dx.doi.org/10.3390/e23101312.

[39] A. Zahid, H. Rashid, M. Shaban, S. Ahmad, E. Ahmed, M. Amjad, M. Baig, M. Arshad, M. Tariq, M. Tariq, M. Zafar, A. Basit, Dynamic S-box design using a novel square polynomial transformation and permutation, IEEE Access 9 (2021) 1–12, http://dx.doi.org/10.1109/ACCESS.2021.3086717.

[40] A. Khompysh, N. Kapalova, K. Algazy, D. Dyusenbayev, K. Sakan, Design of substitution nodes (S-Boxes) of a block cipher intended for preliminary encryption of confidential information, Cogent Eng. 9 (1) (2022) 2080623, http://dx.doi.org/10.1080/23311916.2022.2080623, arXiv:https://doi.org/10.1080/23311916.2022.2080623.

[41] L.E. Bassham III, A.L. Rukhin, J. Soto, J.R. Nechvatal, M.E. Smid, E.B. Barker, S.D. Leigh, M. Levenson, M. Vangel, D.L. Banks, N.A. Heckert, J.F. Dray, S. Vo, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Tech. Rep., National Institute of Standards and Technology, Gaithersburg, MD, 2010.

[42] D. Apdilah, M.K. Harahap, N. Khairina, A.M. Husein, M. Harahap, A comparison of one time pad random key generation using linear congruential generator and quadratic congruential generator, J. Phys.: Conf. Ser. 1007 (1) (2018) 012006, http://dx.doi.org/10.1088/1742-6596/1007/1/012006.

[43] T. Lugrin, One-Time Pad, Springer Nature Switzerland, Cham, 2023, pp. 3–6, http://dx.doi.org/10.1007/978-3-031-33386-6_1.

[44] J.W. Jolles, Broad-scale applications of the Raspberry Pi: A review and guide for biologists, Methods Ecol. Evol. (2021) http://dx.doi.org/10.1111/2041-210X.13652.

[45] C. Intel, Intel$^{®}$ NUC, 2021, URL https://simplynuc.com/wp-content/uploads/2021/01/NUC11TN_TechProdSpec.pdf. (Accessed 06 March 2023).

[46] Pixabay Webpage, 2023, URL https://pixabay.com/videos/. (Accessed 11 June 2023).

[47] Motchallenge Webpage, 2023, URL https://motchallenge.net/data/MOT16/. (Accessed 20 August 2023).

[48] Pexels Webpage, 2023, URL https://www.pexels.com/. (Accessed 17 July 2023).

[49] N.T. Courtois, M. Georgiou, M. Scarlata, Slide attacks and LC-weak keys in T-310, Cryptol. 43 (3) (2019) 175–189, http://dx.doi.org/10.1080/01611194.2018.1548392, arXiv:https://doi.org/10.1080/01611194.2018.1548392.

[50] T. Bentahar, Number of pixel change rate and unified average changing intensity for sensitivity analysis of encrypted inSAR interferogram, Ing. Des Syst. D Inf. 25 (2020) 601–607, http://dx.doi.org/10.18280/isi.250507.

[51] Y.M. Al-Moliki, M.T. Alresheedi, Y. Al-Harthi, A.H. Alqahtani, Robust lightweight-channel-independent OFDM-based encryption method for VLC-IoT networks, IEEE Internet Things J. 9 (6) (2022) 4661–4676, http://dx.doi.org/10.1109/JIOT.2021.3107395.

[52] J. Cheng, S. Guo, J. He, An extended Type-1 generalized feistel networks: Lightweight block cipher for IoT, IEEE Internet Things J. 9 (13) (2022) 11408–11421, http://dx.doi.org/10.1109/JIOT.2021.3126317.

[53] Q. Lai, G. Hu, U. Erkan, A. Toktas, A novel pixel-split image encryption scheme based on 2D Salomon map, Expert Syst. Appl. 213 (2023) 118845, http://dx.doi.org/10.1016/j.eswa.2022.118845, URL https://www.sciencedirect.com/science/article/pii/S0957417422018632.

[54] I. Aribilola, B. Lee, M.N. Asghar, Pixel tampering detection in encrypted surveillance videos on resource-constrained devices, Internet Things 25 (2024) 101058, http://dx.doi.org/10.1016/j.iot.2023.101058, URL https://www.sciencedirect.com/science/article/pii/S2542660523003815.

[55] K. Kimani, V. Oduol, K. Langat, Cyber security challenges for IoT-based smart grid networks, Int. J. Crit. Infrastruct. Prot. 25 (2019) 36–49, http://dx.doi.org/10.1016/j.ijcip.2019.01.001, URL https://www.sciencedirect.com/science/article/pii/S1874548217301622.