IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 18, 2021, accepted March 23, 2021, date of publication March 26, 2021, date of current version April 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068931

A Novel Parallelizable Chaotic Image Encryption Scheme Based on Elliptic Curves

ALAA M. ABBAS[1,2], AYMAN A. ALHARBI[3], AND SALEH IBRAHIM[1,4]

[1]Electrical Engineering Department, College of Engineering, Taif University, Taif 21974, Saudi Arabia
[2]Department of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt
[3]Department of Computer Engineering, Umm Al-Qura University, Mecca 21955, Saudi Arabia
[4]Department of Computer Engineering, Faculty of Engineering, Cairo University, Giza 12613, Egypt

Corresponding author: Saleh Ibrahim (saleh@eng.cu.edu.eg)

ABSTRACT Most of the computational time in many chaotic image encryption schemes is spent generating the required chaotic sequences. Since chaotic systems are defined by recurrence relations, they are often generated sequentially. In this paper, we propose a chaotic image encryption scheme which enables pixel-level parallelism to boost the computational speed of generating chaotic sequences. We use a group defined over elliptic curve (EC) points and the addition operator to generate a discrete chaotic sequence and use it to construct an image encryption scheme. The proposed scheme is designed such that encryption and decryption operations are highly parallelizable to take advantage of readily available parallel processing platforms such as GPU acceleration, DSPs and multi-core CPUs. Complexity analysis indicates that the proposed scheme is more efficient than existing EC-based image encryption schemes. Practical experiments on a quad-core CPU show that the proposed scheme can achieve a speedup of 3.93, confirming its superior parallelization efficiency in comparison with existing parallel image encryption schemes. We also provide detailed analysis of the immunity of the proposed scheme to all common cryptanalysis attacks. Results reveal that the proposed technique shows promising performance in terms of security and efficiency.

INDEX TERMS Image encryption, parallel processing, elliptic curve, chaotic maps.

I. INTRODUCTION

Protecting privacy and confidentiality in the age of the connected world is of paramount significance. Due to the increased usage of digital cameras, private digital images must be protected using encryption during storage and transmission.

In addition to their security requirements, image encryption techniques must have high throughput, that is, they must be capable of processing large amounts of data in a short time. Because of the availability of parallel processing capabilities on most computing platforms, parallel processing is an attractive approach to improve image encryption throughput. However, most image encryption schemes are not designed to leverage such hardware capabilities. Specifically, most image encryption schemes depend on the computation of chaotic sequences, which are usually defined by recurrence relations that must be computed sequentially.

A major motivation of this work is to overcome this problem by designing a fully parallelizable image encryption scheme. To achieve this, we turn to elliptic curves and specifically to the point addition and multiplication operations as an alternative source of entropy rather than traditional chaotic maps. We propose an image encryption scheme, in which the chaotic sequence can be generated by performing EC arithmetic operations in parallel. Therefore, the proposed scheme can exploit existing hardware resources, such as multicore CPUs and Single-Instruction-Multiple-Data (SIMD) hardware, e.g., graphics processing units (GPUs) and digital signal processors (DSPs), to speed up computation and increase encryption throughput. Another crucial objective of this work is to guarantee that the proposed scheme is secure against chosen-plaintext attacks (CPA), as well as other common cryptanalysis attacks.

The contribution of the paper can be summarized as follows:

- The proposed image encryption scheme is composed of fully parallelizable components to overcome the limitation of traditional sequential chaotic maps and increase encryption and decryption throughput,
- The proposed scheme facilitates efficient partial image decryption, that is, any arbitrary part of the encrypted

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

image can be decrypted without having to decrypt the entire cipher image, and
- The proposed scheme exhibits enhanced resistance against various attacks including CPA.

The rest of the paper is organized as follows. Section 2 describes related work and covers the necessary background. Section 3 demonstrates the proposed image encryption scheme and its parallel implementation. In Section 4, the security and efficiency analysis of the proposed scheme is presented and compared with recent and relevant schemes. The conclusion is presented in Section 5.

II. BACKGROUND AND RELATED WORK

In this section, we first give a necessary background on chaotic image encryption and review relevant schemes utilizing parallel processing. Then, we review finite elliptic curves and its application in pseudorandom number generation. Finally, we review dynamic S-box construction methods.

A. CHAOTIC IMAGE ENCRYPTION

The unvarying nature of a block cipher transformation makes it unsuitable for directly encrypting highly correlated image data. One common approach to overcome this limitation is to exploit the ergodicity of chaotic map to generate a one-time pad, which adds randomness to plain images [1]. Another important property of chaotic maps is their high sensitivity to initial conditions, which allows the initial conditions to serve as encryption keys [1]. Therefore, chaotic maps emerged as a viable alternative for encrypting image data and the area attracted more research in the past few years. Numerous chaotic maps have been proposed in the literature and used in image encryption schemes. In general, a chaotic map operates on plain image pixels by altering their coordinates (scrambling) or their values (masking), to achieve the confusion required for encryption.

Chaotic scrambling of pixel locations is strongly effective in canceling the high spatial correlation of images. Therefore, chaotic scrambling is part of many image encryption schemes such as [2]–[5]. However, scrambling is not sufficiently secure because it does not affect cipher image histograms. The histogram of a cipher image must not reveal any statistically significant information about the plain image. Therefore, scrambling must be supplemented by other confusion techniques. One such technique is to use a chaotic map to generate a pseudorandom sequence which is XORed with image pixels to mask their values such that the resulting cipher image always has a uniform histogram. In [2], for instance, the chaotic scrambling, performed using an Arnold cat map, is supplemented by chaotic masking using a logistic map. The authors of [3] supplemented the bit shuffling within each block of image pixels with chaotic masking using a combination of three chaotic maps to generate the random mask. However, the computation of both the complex chaotic map and the bit shuffling process takes a relatively high time. Similarly, the schemes proposed in [4] and [5] combine chaotic scrambling and chaotic masking to achieve secure image encryption, but at a high computational cost.

Several image encryption schemes depend on chaotic masking without scrambling such as [6]–[11]. The scheme proposed in [6] generates a chaotic sequence to XOR with plain image pixels and then applies a dynamic S-box substitution to produce the cipher image. In [7], the authors used a Chen chaotic system to construct a group of dynamic S-boxes. Then for each plain image pixel, an S-box is selected at random from the group and applied to obtain the corresponding cipher pixel. The authors in [8] proposed to use a random S-box from three constructed S-boxes for pixel substitution. A chaotic map is used for randomly selecting the S-box to be applied to each pixel. The authors in [9] used a logistic map to generate a chaotic sequence, which is XORed with cipher image pixels. A substitution using S-boxes is applied to the plain pixels before chaotic masking. Evidently, the computational speed of chaotic maps is a crucial factor in determining the throughput of a chaotic encryption scheme. The recent work of [10] studies the performance of several classical and modern chaotic maps for image encryption and compares their computational efficiency. The authors introduced an image encryption framework with provable security characteristics by combining chaotic masking and dynamic substitution boxes. Similarly, the work of [11] proposed a secure image encryption scheme based on Henon map because of its high computational speed. However, all chaotic maps used in these works are sequentially computed. This limitation means that these image encryption schemes cannot exploit parallel processing hardware when available.

B. IMAGE ENCRYPTION SCHEMES UTILIZING PRALLEL PROCESSING

Chaotic image encryption schemes can be designed to benefit from parallel processing in various ways. Several image encryption schemes utilize parallel processing to process multiple images in parallel. For instance, the work proposed in [12] encrypts multiple images in parallel using an encryption pipeline to maximize the utilization of available processor resources. The scheme proposed in [13] processes multiple frames in parallel. Each frame is further split into blocks, and a Baker map is used for local shuffling and a Chen map is used for global shuffling. However, the definition of the encryption key is not clear and CPA analysis is missing.

A common way to utilize parallelism is to divide a plain image into blocks and process these blocks in parallel. For instance, the scheme proposed in [14] divides the image into equally sized blocks and encrypts each block separately in parallel. A separate logistic map is initialized for each block. The initial conditions of the block logistic map are derived from a master image-wide logistic map.

In [15], an encryption context is first derived from the secret key and then reused to encrypt each of the image blocks in parallel. In [16], a 16×16-pixel chaotic window is generated and reused to encrypt plain image blocks in parallel. The authors of [17] proposed a parallel diffusion

method, which iterates a number of parallel coupled map logistic lattices to diffuse plain image blocks in parallel. To reduce the time required for generating an $m \times n$-long chaotic sequence, the scheme proposed by [18] first generates two chaotic sequences of length n each, and applies a linear transformation on each pair of states to derive the control parameters of one of the m logistic maps. Then, each one of the m logistic maps generates a chaotic sequence of length n.

Some parallel image encryption schemes divide the image into groups of rows. The scheme proposed in [19] assigns a group of rows for each processing element, which is then substituted, diffused and permutated locally by one processing element. Subsequently, pixels are permutated across groups to achieve higher diffusion. An alternative approach proposed in [20] iterates an independent chaotic map by each processor. A master key-dependent chaotic map generates the initial conditions for every secondary chaotic map.

The scheme proposed in [21] splits the image into groups of rows, one for each thread, and initializes a chaotic map with an individual initialization for each thread. The chaotic map is used both for shuffling and masking of the pixels. Consequently, the shuffling strength of the algorithm is reduced when multiple threads are used. Similarly, the scheme proposed in [22] divides the image among available processing units to encrypt, and then combines the cipher blocks into a cipher image. Since the diffusion process is performed within each block separately, there exists a tradeoff between diffusion strength and computational efficiency. Moreover, the generation of the chaotic sequence is performed sequentially before the plain image is divided into blocks, thus limiting the achievable speedup. To fairly evaluate the speedup of both algorithms [21], [22], the number of partitions must be the same in the sequential case and in the parallel case such that the confusion strength is the same in both cases.

The scheme proposed in [23] processes the rows of the image individually in parallel, then processes the columns individually in parallel. However, the scheme needs to generate chaotic sequences of size equal to the image, which must be performed sequentially. The authors provided no efficiency analysis to support their claims about the speed of their scheme.

Another approach to the parallelization of image encryption schemes was given in [24]. The plain image is decomposed into bit planes, each of which can be processed independently in parallel. The scheme used a genetic algorithm-like procedure to permute pixels across rows and across columns.

The highest possible level of parallelism is pixel-level parallelism, in which each plain image pixel can be processed independently. However, this is difficult to achieve in chaotic image encryption because of the sequential nature of chaotic map iteration. The scheme of [25] uses a Game of Life method to confuse the plain image. However, the Game of Life is supported by a piecewise linear chaotic map, which must be calculated sequentially. The scheme recently proposed in [26] first scrambles the image using a chaotic map and then applies a cellular automaton rule multiple times to improve diffusion. However, the number of iterations of the chaotic map is linear in the image size, thus defying parallelization.

Several parallel image encryption schemes suffer from chosen-plaintext attack vulnerability, because the chaotic keystream is not image-dependent. For instance, the authors of [27] demonstrated how to break a parallel image encryption scheme using simple CPA.

C. ELLIPTIC CURVE APPLICATIONS IN IMAGE ENCRYPTION

Image encryption schemes can use elliptic curves for several purposes such as (a) pseudorandom number generation (PRNG) [28], [29], (b) for key exchange [2], [30], [31], and (c) encryption of pixels [32], [33].

The system proposed by [28] uses EC to generate a pseudorandom number mask. To reduce the number of point addition, their system uses the AES to expand a short random sequence to match the image size. However, this performance improvement comes at the cost of security. The authors in [29] presented another pseudorandom generator based on EC. Their method needs to generate and sort all points of the EC, which is computationally expensive.

In [2], elliptic curve cryptography is used for exchanging the image encryption parameters controlling the initial conditions of a chaotic sequence and the number of scrambling rounds. The scheme proposed in [30] uses EC cryptosystem to establish shared keys. In [31], the EC Diffie-Hellman exchange is used for key agreement.

The system proposed in [32] embeds scrambled pixel values into EC points and uses ElGamal encryption to generate the cipher image. However, the same private key is reused for encrypting many points causing serious security flaw. The authors of [33] used Koblitz encoding to embed plain image pixel into an EC. The EC points are then encrypted using ElGamal cryptosystem.

D. ELLIPTIC CURVE PRNGS

In this subsection, we introduce the definition of an elliptic curve defined over a prime field and review EC-based pseudorandom number generators.

The elliptic curve defined on a finite field has several applications in cryptography. More specifically, given a finite field $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, where p is a prime number, and two parameters a, and $b \in \mathbb{Z}_p$, where $4a^3 + 27b^2 \neq 0$, an elliptic curve $E(p, a, b)$ is defined as the union of the set of points $(x, y) \in \mathbb{Z}_p^2$, such that $y^2 = x^3 + ax + b$, and the point at infinity, O.

The points of the elliptic curve form an abelian group, where the sum of any two points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ is the point, $T = (x_T, y_T)$, on the curve. The addition

operator is defined as follows:

$$P+Q=\begin{cases}P, & Q=O\\ Q, & P=O\\ O, & x_P=x_Q,\ y_P=p-y_Q\\ (x_T,Y_T), & \text{otherwise}\end{cases} \tag{1}$$

where

$$x_T=\lambda^2-x_P-x_Q\ (\text{mod}\ p),$$
$$y_T=\lambda\left(x_P-x_T\right)-y_P\ (\text{mod}\ p),$$

and

$$\lambda=\begin{cases}(3x_P^2+a)/2y_P, & P=Q\\ (y_P-y_Q)/(x_P-x_Q) & \text{otherwise}\end{cases}$$

Scalar multiplication is defined as the repeated addition of a point to itself, that is,

$$kP=\underbrace{P+P+\ldots+P.}_{k\ times}$$

In practice, point multiplication is usually performed more efficiently using the double-and-add multiplication algorithm [34]. A variation of the multiplication algorithm due to Montgomery [35] adapts the structure of double-and-add to make its execution time constant regardless of the value of the factor, $k\in\mathbb{Z}_p$, to resist side channel timing and power analysis attacks.

Several EC-based PRNGs can be found in the literature because of the chaotic behavior caused by point addition and multiplication. The PRNG in [36], uses an additive group defined by a generator point $G\in E\left(p,a,b\right)$. A seed, s, is multiplied by G and the most significant bit (MSB) of the product point, sG, is output as a pseudorandom bit. The generator, G, is successively added to the current point and the MSBs are appended to the output pseudorandom bit sequence, $\langle \text{MSB}\left(\left(s+i\right)G\right)\rangle_i$. A more general EC-based construction analogous to the linear congruential generator, $P_i=aP_{i-1}+B$, with output sequence $\left\langle \log_G X_i\right\rangle$, was studied in [37] and found not to satisfy the definition of a cryptographically secure PRNG (CSPRNG). An alternative construction was proposed in [37], in which the state update function is $P_i=y\left(P_{i-1}\right)G$, and the output bit sequence is $\langle \text{LSB}(y\left(P_i\right))\rangle$. The new PRNG was shown to be cryptographically secure based on the hardness of the EC discrete logarithm problem (ECDLP).

The PRNG proposed in [38] starts from a seed u_1 and iterates the state update function, $u_{i+1}=x\left(u_iG\right)+i$, where $G\in E\left(\mathbb{Z}_p\right)$ is a generator point. The output is the x-coordinate of the underlying point sequence, that is, $\langle x(u_iG)\rangle$. The period of this PRNG is equal to the size of the prime field, p.

In [39], a power generators based on EC was proposed, where $P_i=e^iG$, for some characteristic $e\geq 2$ and a generator $G\in E\left(\mathbb{Z}_p\right)$. This PRNG was shown to have a good statistical distribution with sufficiently large period.

The dual EC pseudorandom bit generator (DRBG) proposed in [40], uses two one EC point, $G\in E\left(\mathbb{Z}_p\right)$, to generate a point sequence and another point $Q\in E\left(\mathbb{Z}_p\right)$ to generate the output. Specifically, $u_i=x\left(u_{i-1}G\right)$ and $r_i=x(u_iQ)$. The two points G and Q must be randomly chosen and enough output bits must be truncated from r_i for DBRG to be considered secure [41]. Apart from the variants of the linear congruential generator, EC-based PRNGs are difficult to parallelize.

E. DYNAMIC S-BOXES

An S-box is a function that substitutes each input bit pattern with a corresponding output bit pattern. Dynamic S-boxes have a mapping that varies from one encryption session to the next, thus avoiding classical linear, differential, and algebraic cryptanalysis applicable to static S-boxes. The use of dynamic key-dependent S-boxes adds another layer of confusion, which increases key space and improves security of image encryption schemes [42]. The framework of [10] is a recent example of image encryption employing dynamic S-boxes. Several dynamic S-box construction methods can be found in the literature. The general outline of such construction methods is based on a PRNG. The generated sequence of random numbers is processed by some algorithm to produce a corresponding bijective S-box. In [11], for instance, the algorithm starts with an initial bijective S-box and uses the generated sequence of pseudorandom numbers to iteratively swap elements of the initial S-box to construct the final S-box. In [10], the algorithm filters repeated numbers out of the sequence to produce a sequence of unique random numbers that represent the final S-box. In [43], a Fisher-Yates shuffle modifies the initial S-box to generate a uniformly distributed random S-box. The latter method has the advantage of proven resistance to chosen-plaintext attacks.

III. PROPOSED IMAGE ENCRYPTION SCHEME

The proposed encryption scheme performs two main operations. First, we XOR the plain image with a pseudorandom keystream. Then, the result is substituted using a dynamic key-dependent S-Box.

At the core of the proposed image encryption scheme is the EC-based linear congruential generator (LCG) PRNG. For a high order generator point $G\in E\left(\mathbb{Z}_p\right)$, the x-coordinates of the sequence of points $\langle iG\rangle_{i=1}^{L}$, exhibit chaotic properties [44]. Specifically, the sequence exhibits ergodicity and high sensitivity to G. In contrast to the elements of a traditional chaotic maps, which must be calculated sequentially using a state update rule, the elements of the sequence $\langle x\left(iG\right)\rangle_{i=1}^{L}$ can be calculated in parallel using efficient double-and-add EC point multiplication operations. Therefore, the sequence $\langle x\left(iG\right)\rangle_{i=1}^{L}$ will be used for image encryption.

A plain image is represented in lexicographic order as a sequence of pixels, $\langle I_i\rangle_{i=1}^{L}$, where L is the image size. The sending and the receiving parties share a composite secret key (k_S,k_C) as well as a publicly known EC generator point, G, of a large prime order. During the initialization phase, the two

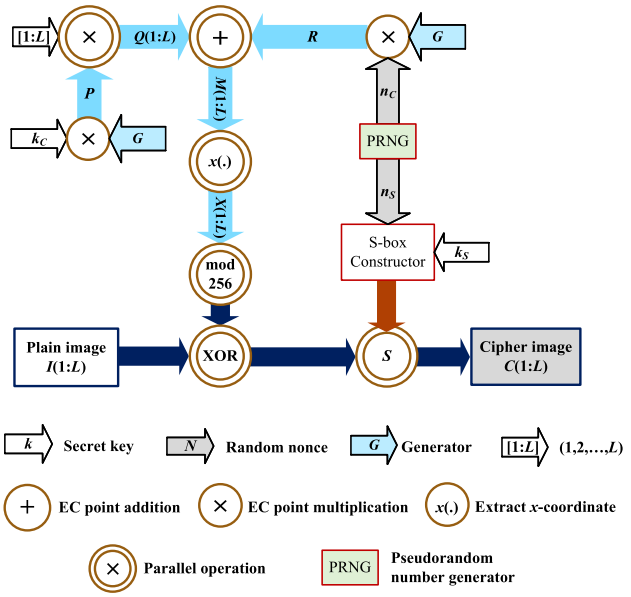

FIGURE 1. Encryption block diagram.

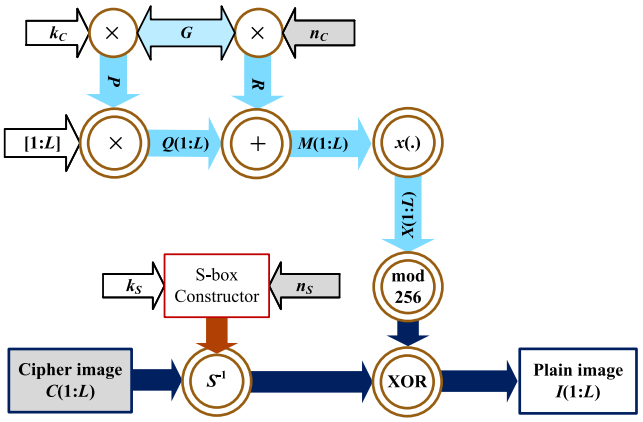

FIGURE 2. Decryption block diagram.

parties use k_C to construct an initial EC point sequence, $\langle Q_i\rangle_{i=1}^{L} = \langle ik_C G\rangle_{i=1}^{L}$. This sequence will be reused each time an image is to be encrypted or decrypted.

The encryption scheme block diagram is shown in Figure 1. To encrypt a plain image, the sender uses a PRNG to generate a pair of image-dependent nonce values, (n_S, n_C). The first nonce value, n_S, is used along with the S-box key, k_S, to construct a dynamic key-dependent S-box, S, using any S-box construction method, which satisfies the requirements of efficiency, dynamicity and key-dependence [10]. The other nonce value, n_C, is converted to a random EC point, $R = n_c G$, which is added in parallel to the initial point sequence to generate a modified point sequence, $\langle M_i\rangle_{i=1}^{L} = \langle R + Q_i\rangle_{i=1}^{L}$. The x-coordinate of each point of the sequence $\langle M_i\rangle_{i=1}^{L}$ is converted in parallel to a pixel via a modulo 256 operation to obtain a keystream. The keystream is XORed in parallel with the plain image pixels. Then, the dynamic S-box, S, is applied in parallel to the masked pixels to produce the cipher image pixels, $\langle C_i\rangle_{i=1}^{L}$. The cipher message is composed of the two nonce values and the cipher image pixels.

As shown in Figure 2, the receiver decrypts a cipher image by first extracting the nonce values from the cipher message. The receiver uses the first nonce, n_S, along with the secret S-box key, k_S, to construct the same dynamic S-box, S, that was used for encryption. The S-box is then inverted to obtain, S^{-1}, which is used for decryption. The other nonce, n_C, modulates the initial point sequence, $\langle Q_i\rangle_{i=1}^{L}$, to obtain the same keystream used for encryption, $\langle M_i\rangle_{i=1}^{L} = \langle n_C G + Q_i\rangle_{i=1}^{L}$. After applying S^{-1} to the cipher-image pixels, the result is XORed with $\langle M_i\rangle_{i=1}^{L}$ to obtain the decrypted image.

It is worth noting that each of the encryption and decryption operations that depend on the image size can be parallelized to take advantage of SIMD computing architecture thus reducing the image encryption and decryption time. Namely, the addition of the random point R to the sequence $\langle Q_i\rangle_{i=1}^{L}$, the extraction of the keystream from the point sequence, the pixel-mask XOR operation, and the S-box substitution can each be performed on pixels in parallel.

The initial point sequence $\langle Q_i\rangle_{i=1}^{L} = \langle ik_C G\rangle_{i=1}^{L}$ can also be calculated in parallel in a straightforward manner using one multiplication per point. Since EC point multiplication is often orders-of-magnitude slower than point addition, it is desirable to design a more efficient parallel algorithm for calculating $\langle Q_i\rangle_{i=1}^{L}$. The recursive definition of the point sequence, $Q_{i+1} = Q_i + k_C G$, requires only one initial multiplication to obtain $P = k_C G$. Subsequently, the entire sequence of points can be obtained using just one addition per point. However, the recursive definition does not lend itself to parallelization. Therefore, we propose the following method to exploit m parallel processors to generate the sequence, $\langle Q_i\rangle_{i=1}^{L}$.

First calculate $P = k_C G$. Then, the point sequence $\langle Q_i\rangle_{i=1}^{L}$ is divided into m subsequences $\langle Q_i\rangle_{i=1}^{l}$, $\langle Q_i\rangle_{i=l+1}^{2l}$, $\ldots$, $\langle Q_i\rangle_{i=(m-1)l+1}^{ml}$, of length $l = \lceil L/m \rceil$ each. The initial point of each subsequence can be calculated in parallel using one EC point multiplication per processor. Namely, Processor $\#j \in \{0, 1, \ldots, m-1\}$ calculates:

$$Q_{jl+1} = (jl + 1)\, P.$$

Subsequently, each processor continues to operate in parallel to compute the remaining points of the corresponding subsequence using one EC point addition per sequence point. Namely, Processor $\#j \in \{0, 1, \ldots, m-1\}$, recursively calculates $Q_{jl+2}, \ldots, Q_{(j+1)l}$ using the formula

$$Q_{jl+i} = Q_{jl+i-1} + P, \forall i \in \{2, \ldots, l\}$$

Therefore, the time complexity of the proposed point sequence generation algorithm on an m-processor parallel computer is $\left(O(\lg p) + O\left(L/m\right)\right)$ EC additions, where p is the EC prime field parameter.

IV. EVALUATION AND COMPARISONS

To evaluate the proposed scheme, we implement the dynamic S-box construction component using the method proposed

in [43], which is based on permutated elliptic curves (PEC). This method has several advantages: (a) the PEC S-box is key-dependent which increases the encryption scheme key-space, (b) the construction method has an additional nonce parameter, which facilitates semantic security, and (c) the method is computationally efficient enabling the construction of a dynamic S-box in less than 1 ms.

In this section, we evaluate two aspects of the proposed image encryption scheme. First, we evaluate the security of the proposed scheme against various attacks including statistical, differential, related key, brute force, and chosen-plaintext attacks. Then, we evaluate the efficiency of the proposed scheme through computational complexity and speedup analysis.

A. STATISTICAL ANALYSIS

An adversary can try to extract some statistical information from the encrypted image. Therefore, the encrypted image must pass traditional statistical tests to overcome the attempts of the adversary to gather information. The following subsections apply traditional statistical tests to guarantee that the proposed scheme is immune against this type of attacks.

1) HISTOGRAM AND CHI-SQUARE VARIANCE (χ^2) TEST

The uniformity of histogram means that all intensity values have equal occurrence probability, which prevents an adversary from guessing any information about the plaintext image. Figure 3 shows a set of standard images, and their corresponding histograms, encrypted images, and encrypted image histograms, which appear uniform.

Since visual inspection of the histogram is not always precise, we resort to the χ^2 variance test to quantitatively measure the uniformity of the encrypted image histograms. The test is performed by calculating the observed histogram variance:

$$\sigma^2 = \frac{1}{l}\sum_{j=0}^{l-1}\left(f_j - \lambda\right)^2 \tag{2}$$

where l is the number of pixel intensity levels, i.e., f_j is the frequency of pixel intensity j in the encrypted image, $0 \le j < l$, and $\lambda = N/l$ is the expectation of f_j [45]. The observed variance is then compared with the hypothesized histogram variance of a uniformly distributed random image, $\sigma_0^2 = \lambda\left(1 - 1/l\right) = \lambda\,(l-1)/l$ using the formula

$$X^2 = (l-1)\left(\frac{\sigma}{\sigma_0}\right)^2 = \frac{(1-l)\left(\frac{1}{l}\sum(f_j-\lambda)^2\right)}{\lambda(l-1)/l} = \sum\frac{\left(f_j-\lambda\right)^2}{\lambda}$$

If image pixel values are uniformly distributed over the l levels, the probability of observing the value of X^2 follows the Chi-squared distribution with $l-1$ degrees of freedom, i.e., χ^2_{l-1}. The cipher image passes the histogram test when the p-value exceeds a certain acceptable significance level, α. To improve test accuracy, we perform the χ^2 test on 100 instances of encrypted images and record the passing rate in Table 1 for $\alpha = 0.01$ and $\alpha = 0.05$. The high pass rates of histogram uniformity tests indicate that the cipher images encrypted with the proposed schemes satisfy the randomness hypothesis.

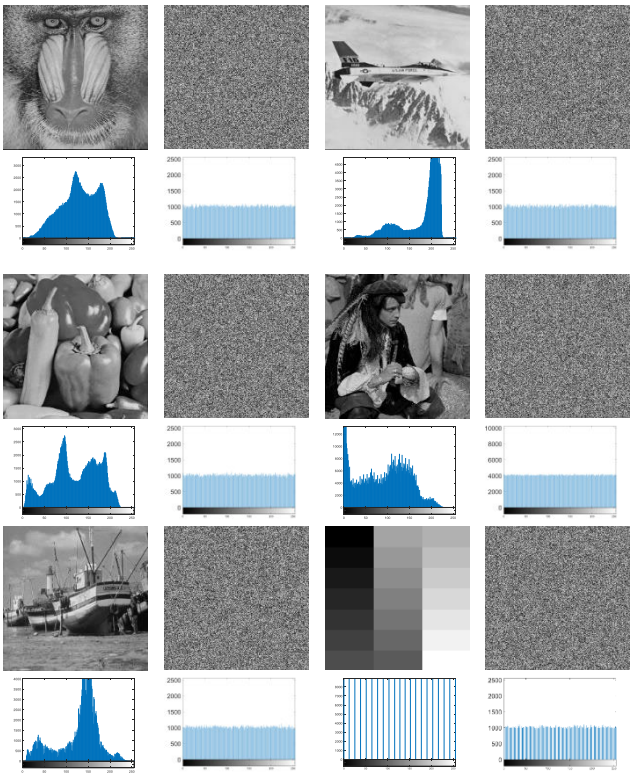

FIGURE 3. Sample images used for statistical testing and their corresponding histogram before and after encryption by the proposed scheme.

TABLE 1. Histogram χ^2 test.

Image	Image size	$\alpha = 0.01$ Pass percent	$\alpha = 0.05$ Pass percent
Mandrill	512×512	100	95
Airplane	512×512	99	97
Pepper	512×512	100	92
Male	1024×1024	99	96
Boat	512×512	98	96
Gray bars	512×512	100	90

2) CORRELATION TESTS

Two correlation tests are useful for evaluating the quality of encryption. The cross-correlation test measures the dissimilarity between the encrypted image and the plaintext image. The spatial auto-correlation test measures the correlation between neighboring pixels in the encrypted image. The cross-correlation coefficient is computed as follows:

$$CC_{m,n} = \frac{cov(m,n)}{\sqrt{v(m)}\cdot\sqrt{v(n)}} \tag{3}$$

TABLE 2. Correlation between plaintext and ciphered images and spatial correlation.

Image	Correlation	Spatial Correlation					
		Horizontal		Vertical		Diagonal	
		Plain	Encrypted	Plain	Encrypted	Plain	Encrypted
Mandrill	0.00241	0.86654	-0.00360	0.75873	-0.00051	0.72618	0.00034
Airplane	0.00054	0.96631	-0.00244	0.96412	-0.00301	0.93702	-0.00024
Pepper	-0.00300	0.97677	0.00119	0.97920	-0.00166	0.96393	0.00180
Male	-0.00031	0.97744	0.00074	0.98126	-0.00116	0.96714	-0.00140
Boat	0.00282	0.93811	0.00027	0.97131	0.00094	0.92216	0.00190
Gray bars	0.00005	0.99654	0.00298	0.99983	0.00223	0.99639	-0.00217

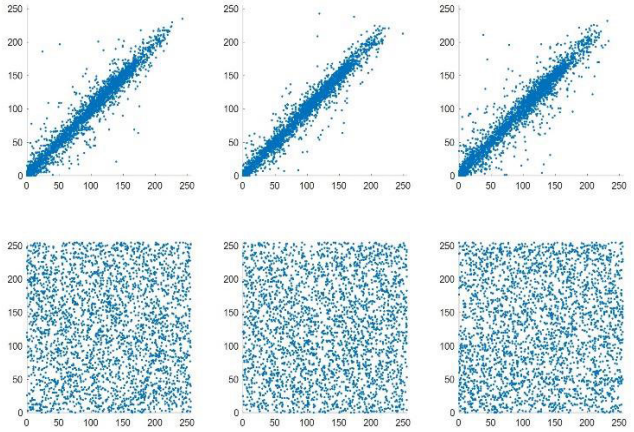

FIGURE 4. Spatial correlation distribution of the male image of the plain image (top row) and its corresponding cipher image (bottom row) in the Horizontal (left), Vertical (middle) and Diagonal (right) directions.

where m and n are the plaintext image and the encrypted image, respectively, and

$$cov\,(m,n) = \frac{1}{N}\sum_{j=1}^{N}\left(m_j-\bar{m}\right)\left(n_j-\bar{n}\right),$$
$$v\,(m) = \frac{1}{N}\sum_{j=1}^{N}(m_j-\bar{m})^2,$$
$$v\,(n) = \frac{1}{N}\sum_{j=1}^{N}(n_j-\bar{n})^2,$$
$$\bar{m} = \frac{1}{N}\sum_{j=1}^{N} m_j,\ and\ \bar{n} = \frac{1}{N}\sum_{j=1}^{N} n_j,$$

where N is the number pixels in the image.

The spatial correlation coefficients of an image m of width W and height H can be computed, as follows:

$$AC\,(dx,dy) = \frac{scov\,(m,dx,xy)}{v\,(m)},$$

where $scov\,(m,dx,dy) = (1/\mathrm{N})\sum_{x=1}^{W-dx}\sum_{y=1}^{H-dy}$ $(m\,(x,y)-\bar{m})\,(m(x+dx,y+dy)-\bar{m})$. For horizontal, vertical, and diagonal correlations we take (dx, dy) to be $(1, 0)$, $(0, 1)$, and $(1, 1)$ respectively.

Table 2 shows the correlation and spatial correlation values of the tested images. The results indicate that the proposed scheme successfully breaks the correlation between the plaintext image and the encrypted image and breaks the spatial correlation. Figure 4 illustrates the spatial correlation of the encrypted image in contrast with the plaintext image.

TABLE 3. The entropy test values of the plaintext image and the ciphered image.

Image	Entropy	
	Plain	Encrypted
Mandrill	7.358337	7.999359
Airplane	6.702463	7.999377
Pepper	7.593654	7.999298
Male	7.523737	7.999813
Boat	7.19137	7.999385
Gray bars	4.392295	7.999408

The spreading of pixel intensity cooccurrence shows that the neighboring pixels of the encrypted image are uncorrelated.

3) ENTROPY TEST

The entropy test measures the randomness of an image. The encrypted image must gain a score of an entropy near 8 as possible to guarantee that the image is as close as a random image. The entropy of image, m, is conducted as follows:

$$E\,(m) = -\sum_{j=0}^{255} P_j\,log_2\,P_j, \tag{4}$$

where P_j is the probability of occurrence of pixel value j.

Table 3 shows the results of entropy of the plaintext image versus the encrypted image. The results of encrypted images are very close to score 8 which indicate that the proposed scheme produces a completely random image.

Table 4 lists entropy and spatial correlation test results of the proposed scheme and recently published related schemes. The results indicate that the proposed scheme has a high performance in comparison with other schemes.

4) NIST RANDOMNESS TESTS

A good cipher image should be statistically indistinguishable from a random sequence. A statistical tests proposed by NIST [46] for pseudorandom number generators can be employed to test the randomness of cipher images.

Table 5 shows the NIST test results for a sample of cipher images resulting from encrypting an all-black image 100 times using the proposed scheme. According to the test suite guide, the minimum passing proportion is 95.89% for the random excursion tests, and 96.41% for all other tests. Consequently, the proposed scheme passes all randomness tests successfully.

TABLE 4. Comparison between the proposed scheme and the related method in terms of entropy and spatial correlation.

Scheme	Entropy	Correlation		
		H	V	D
Proposed	7.9998	0.0007	-0.0012	-0.0014
Ref. **[11]**, 2020	7.9994	0.0021	0.0117	0.0125
Ref. **[10]**, 2020	7.9995	-0.0054	0.0088	-0.0054
Ref. **[47]**, 2020	7.9972	0.0001	−0.0015	0.0013
Ref. **[48]**, 2019	7.9993	-0.0002	-0.0024	0.0013
Ref. **[45]**, 2019	7.9993	0.0013	-0.0049	0.0057
Ref. **[49]**, 2019	7.9975	−0.0022	0.0013	0.0029
Ref. **[50]**, 2019	7.9975	−0.0084	−0.0017	−0.0019
Ref. **[51]**, 2019	7.9988	−0.0058	0.0064	0.0059
Ref. **[31]**, 2018	7.9986	0.0012	0.0044	0.0046
Ref. **[52]**, 2017	7.9994	0.0035	0.0003	0.0034
Ref. **[53]**, 2016	7.9985	-0.0095	-0.0170	-0.0119
Ref. **[54]**, 2015	7.9992	0.0007	-0.0028	-0.0001

TABLE 5. NIST test for encrypted Images.

Test	*p*-Value	Proportion %	Result
Frequency	0.637119	100	Pass
Block frequency	0.657933	99	Pass
Cumulative sums	0.719747	100	Pass
Runs	0.437274	98	Pass
Longest runs of ones	0.514124	97	Pass
Rank	0.289667	99	Pass
Spectral DFT	0.494392	98	Pass
Overlapping template	0.616305	98	Pass
Non-overlap. template	0.262249	100	Pass
Universal	0.040108	98	Pass
Approximate entropy	0.455937	99	Pass
Serial	0.699313	99	Pass
Linear complexity	0.304126	100	Pass
Random excursions	0.505629	98	Pass
Rand. excursion variant	0.327854	100	Pass

5) SECOND ORDER TEXTURE ANALYSIS

Encryption quality can be measured quantitively by analyzing the texture of encrypted images. First, the 256 gray-levels of the encrypted image, C, are divided into 8 equal ranges. The gray-level of each pixel $C(x, y)$ of the of r rows and c columns of C is classified into one of these 8 levels. Then an 8×8 gray-level co-occurrence probability matrix (GLCM) is calculated as follows:

$$p_r(i,j) = \frac{1}{r(c-1)} \# \left\{ (x,y) \mid \left\lceil \frac{C(x,y)}{32} \right\rceil = i, \left\lceil \frac{C(x,y+1)}{32} \right\rceil = j \right\}$$

for all $i, j \in \{1, \ldots, 8\}$. Using the GLCM, second order statistics can be calculated as follows [55].

$$Contrast(C) = \sum_{i,j\in\{1,..,8\}} |\text{i-j}|^2 p_r(i,j) \quad (5)$$

$$Correlation(C) = \sum_{i,j\in\{1,..,8\}} \frac{(i-\mu_i)(j-\mu_j) p_r(i,j)}{\sigma_i \sigma_j} \quad (6)$$

$$Energy(C) = \sum_{i,j\in\{1,..,8\}} p_r(i,j)^2 \quad (7)$$

$$Homogeneity(C) = \sum_{i,j\in\{1,..,8\}} \frac{p_r(i,j)}{1+|\text{i-j}|} \quad (8)$$

An ideal random encrypted image is expected to have a uniform 8×8 gray-level cooccurrence probability matrix, i.e., $p_r(i,j) = 1/64, \forall i, j \in \{1, \ldots, 8\}$. In this case, the optimal contrast, correlation, homogeneity, and energy is 10.5, 0, 0.38940, and 0.015625 [10].

Therefore, high quality encryption should generate cipher images with contrast, correlation, homogeneity, and energy close to the optimal values. Table 6 lists the GLCM-based second order statistics of the encrypted images generated by the proposed scheme. The results indicate the high quality of encryption achieved by the proposed scheme.

B. DIFFERENTIAL ANALYSIS

An encryption system must satisfy the confusion and diffusion requirements for encryption quality. Therefore, an encryption system should respond to a little change in the plaintext image and produce a fully different cipher image. Two numerical values are utilized to measure the ability of the encryption system to resist the differential attacks, the Number of Pixels Change Rate (NPCR), and the Unified Average Change Intensity (UACI). The optimum value of NPCR and UACI are 99.6094% and 33.4635% respectively.

To simulate the differential attack, a plaintext image, p_1, which is encrypted into cipher image, C_1, is modified by changing just one pixel to obtain a plaintext image, p_2, which is encrypted into cipher image, C_2. Then the NCPR and UACI indicators is computed as follows:

$$UACI(C_1, C_2) = \frac{1}{MN} \sum_{i,j} \frac{|C_2(i,j) - C_1(i,j)|}{255} \times 100\%, \quad (9)$$

$$NPCR(C_1, C_2) = \sum_{i,j} \frac{D(i,j)}{MN} \times 100\%, \quad (10)$$

where MN is the number of image pixels, and

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

For more realistic simulation, the experiment is repeated 100 times and the χ^2-test pass rate and the mean are reported. Table 7 shows the results of the correlation between C_1 and C_2, NPCR and UACI. Results indicate that the value of correlation between two encrypted images with only one-pixel value change in the plaintext images is very close to zero which reveal the two images are uncorrelated. Moreover, the high pass rate of the NPCR and the UACI tests indicate that the proposed scheme is immune against differential attacks.

Figure 5 illustrates two encrypted images C_1 and C_2, corresponding to two plaintext images with just one-pixel change. Also, the difference between encrypted images is presented for visual inspection. The difference image is computed as follows:

$$\Delta(i,j) = (C_1(i,j) - C_2(i,j)) \bmod 256. \quad (11)$$

TABLE 6. The majority logic criteria of the encrypted images.

Image	Contrast		Correlation		Homogeneity		Energy	
	Plain	Enc	Plain	Enc	Plain	Enc	Plain	Enc
Mandrill	0.61779	10.4881	0.25290	-0.0004	0.78729	0.38937	0.08895	0.015630
Airplane	0.21208	10.5236	0.74813	0.0010	0.92872	0.38937	0.38085	0.015630
Pepper	0.25607	10.5076	0.47293	-0.0001	0.89893	0.38933	0.11068	0.015629
Male	0.25280	10.5023	0.71718	0.0003	0.89793	0.38939	0.11994	0.015626
Boat	0.37995	10.4288	0.33205	0.0019	0.87530	0.38998	0.18900	0.015629
Gray bars	0.03832	10.4762	0.89419	0.0019	0.99759	0.38988	0.12846	0.015630

TABLE 7. Differential attack test results.

Image	Correlation			NPCR (α=0.01)		UACI (α=0.01)	
	min	mean	max	mean	pass %	mean	pass %
Mandrill	-0.00485	0.000037	0.006575	99.61050	100	33.46436	98
Airplane	-0.00653	0.000059	0.005359	99.60848	98	33.46975	97
Pepper	-0.00409	-0.000150	0.003830	99.60940	99	33.46407	100
Male	-0.00179	-0.000005	0.002138	99.60952	100	33.46213	100
Boat	-0.00364	0.000394	0.006210	99.61046	98	33.45334	99
Gray bars	-0.00401	-0.000097	0.005115	99.60859	98	33.46608	100

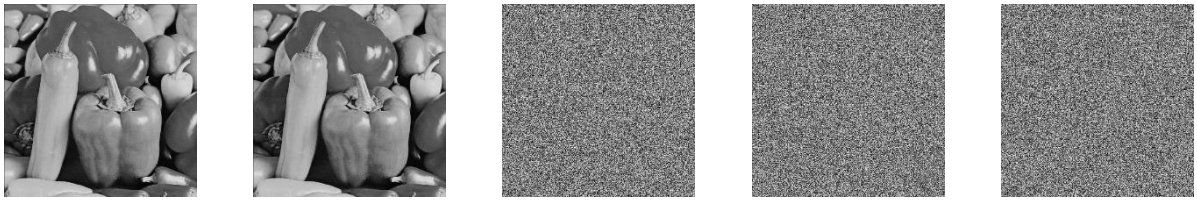

FIGURE 5. Results of plaintext image sensitivity test: (a) original plain image, (b) plain image with one-pixel change, (c) cipher image of (a), (d) cipher image of (b), and (e) difference between (c) and (d).

TABLE 8. Differential attack analysis of the proposed scheme in comparison with relevant schemes.

Scheme	UACI	NPCR
Proposed	33.4621	99.6095
Ref. **[11]**, 2020	33.4409	99.6086
Ref. **[10]**, 2020	33.4644	99.6095
Ref. **[56]**, 2020	33.4590	99.6100
Ref. **[57]**, 2020	33.4100	99.7900
Ref. **[45]**, 2019	33.4121	99.6536
Ref. **[48]**, 2019	33.4200	99.5800
Ref. **[47]**, 2020	24.2534	76.1681
Ref. **[29]**, 2019	33.5000	99.6000
Ref. **[32]** , 2019	33.4682	99.6113
Ref. **[58]** , 2019	33.3797	99.6495
Ref. **[51]** , 2019	33.6259	99.6118
Ref. **[59]**, 2018	33.6046	99.6369
Ref. **[31]** , 2018	33.1100	99.9500
Ref. **[52]**, 2017	33.4423	99.6071
Ref. **[54]**, 2015	33.4682	99.6082

The difference image looks like a random image which indicates that the two encrypted images are completely different.

Table 8 lists the differential analysis comparisons between the proposed scheme and the most recent and relevant schemes. The results indicate that the proposed scheme achieves one of the best values NPCR and UACI test results. In other words, the proposed scheme proved to be very sensitive to plaintext image.

TABLE 9. Keystream key sensitivity analysis, testing each image encrypted with k_C with the same image encrypted with k'_C.

Image	Correlation	NPCR	UACI
Mandrill	0.003959	99.61014	33.33055
Airplane	-0.001000	99.61014	33.49726
Pepper	0.001638	99.61014	33.44339
Male	-0.000056	99.60899	33.47204
Boat	0.000678	99.61014	33.46087
Gray bars	0.002868	99.61014	33.40656

C. KEY SENSITIVITY ANALYSIS

High sensitivity to changes in the key is a desirable property of encryption schemes as it deters related key attacks. To measure the sensitivity of the proposed scheme to changes in the key, we flip the least significant bit of the chaotic mask key, k_C, and test the resulting difference in cipher image. Table 9 shows the correlation, NPCR and UACI test results between a pair of cipher images corresponding to a change in the chaotic mask key from k_C to k'_C, where k_C = "515114605 267721723715377806812181419767246164712119438895 715414582411936817033968026396318892210121150039 6 89898465184126 159556515674168168785592140950526 1," and $k'_C = k_C \oplus 1$.

Figure 6 illustrates the high sensitivity of the proposed encryption scheme to changes in the k_C component of the encryption key and the scheme immunity to related key attacks on k_C.

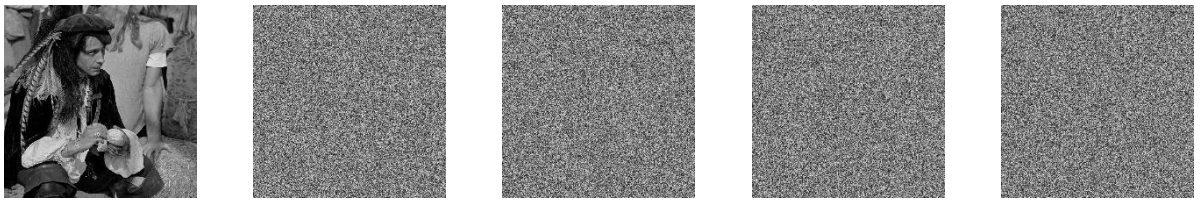

FIGURE 6. The chaotic key sensitivity test for male image. (a) original plain image, (b) cipher image encrypted with K_C, (c) cipher image encrypted with K_C', (d) difference between cipher images encrypted with K_C and K_C', (e) image decrypted with K_C' after encryption with K_C.

TABLE 10. Analysis of sensitivity to random nonce, N_C, to demonstrate immunity to chosen-plaintext attacks. n_C = 5114891825938631676842962321799348629396326253107130922636763261917 339240650587896535510156660908782813251634742791178539403524505596745655150613615482 40 n_C' = 5114891825938631676842962321799348629396326253107130922636763261917339240650587896535510156660908782813251634742791178539403524505596745655 15061361548241.

Image	Correlation	UACI (α=0.05)			NPCR (α=0.05)		
			p-Value	Result		*p*-Value	Result
All-White	0.000414	33.49555	0.488465	Pass	99.620819	0.826221	Pass
All-Black	0.001398	33.44081	0.622855	Pass	99.620819	0.826221	Pass
Gray bars	-0.003478	33.51990	0.222599	Pass	99.620819	0.826221	Pass

D. RESISTANCE TO CHOSEN-PLAINTEXT ATTACKS

In a chosen-plaintext attack, the adversary encrypts a chosen plaintext and attempts to use the information obtained from the corresponding ciphertext to facilitate the decryption of other ciphertexts encrypted using the same key.

The proposed image encryption scheme has a component of randomness due to the S-box random nonce, n_S, and the chaotic mask random nonce, n_C. Therefore, each time the same plain image, I, is encrypted using the same key pair, (k_C, k_S), a different cipher image is produced. To demonstrate how the probabilistic nature of the proposed image encryption scheme hinders chosen-plaintext attacks, we encrypt an all-black image twice using the same key while varying only the random nonce, n_C. The resulting two cipher images are compared using cross-correlation, NPCR and UACI randomness measures. Results in Table 10 show that a slight change in the nonce, n_C, results in a large change in the cipher image as evident by the value of NPCR. The correlation and UACI values show that two cipher images are uncorrelated. Figure 7 shows the original all-black image, I, the cipher image, C, resulting from encryption with nonce, n_C, the cipher image, C', resulting from encryption with a slightly different nonce, n_c', and the difference image $\Delta(C, C')$ using (11).This result shows that the proposed scheme is highly sensitive to the nonce, n_C. Therefore, the information obtained by encrypting a chosen plain image is statistically insignificant.

E. KEY SPACE ANALYSIS

The secret key of the proposed image encryption scheme is composed of two components: k_C, which controls the keystream, and k_S which controls the construction of the S-box. The total effective key space is thus equal to the product of the two effective key spaces of k_C and k_S. The key space of k_C is equal to the order of the generator G, which depends on the selected EC. In case of the standard Curve25519, the order of G is 2^{252}. Theoretically, the space of dynamic S-boxes is $2^n \cong 2^{1684}$. However, the effective key space of the dynamic S-box is limited in practice by the size of k_S and the properties of the S-box construction method. When the EC-based S-box construction method of [43] is used with Curve25519, the S-box key space of k_S is approximately 2^{255}. Consequently, the overall key space of the proposed scheme is 2^{507} which is well beyond the threat of brute-force attacks.

TABLE 11. Computational complexity comparison.

Scheme	Time complexity (EC point additions)
Proposed	$O(L/m)$
Ref. [28] *	$O(\alpha \lg p)$
Ref. [29] **	$O(p/m)$
Ref. [2]	$O(8L/m)$
Ref. [33]	$O((L \lg p)/m)$

*α is the length of the generated EC pseudorandom sequence. Security requires $\alpha = L/16$.

** Security requires that $p \gg L$

F. SPEED ANALYSIS

1) THEORETICAL SPEED ANALYSIS

The proposed encryption scheme spends most of the time generating the keystream. To reduce the encryption time per image, the initial point sequence, $\langle Q_i \rangle_{i=1}^{L}$, is computed only once for a given secret key, k_C, and then $\langle Q_i \rangle$ is cached and reused for each subsequence image. This reduces the image encryption workload to just L point additions per image, where L is the image size in pixels. In Table 11, we compare the computational complexity of the proposed image encryption scheme with related EC-based image encryption schemes in both the sequential and the parallel processing cases. Figure 8 illustrates the theoretical time complexity of the proposed scheme using parallel processing in comparison

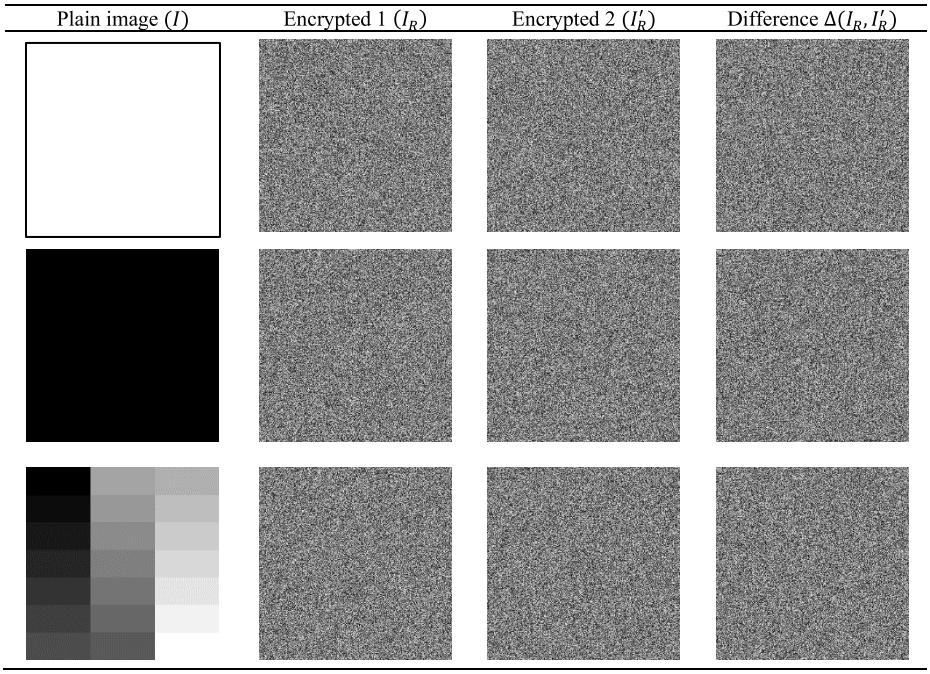

FIGURE 7. Chosen plaintext attack analysis using all-white image (top), all-black image (middle), and gray bars image (bottom), showing original plain image (I), encrypted original image (I_R), encrypted changed image (I'_R) and the difference between encrypted images.

with related schemes. The points corresponding to each scheme indicate the number of per-processor EC point additions required to encrypt a 1024 × 1024 grayscale image using a 256-bit EC at a given number of parallel processors, m. The points corresponding to the scheme of [28] indicate that the scheme is not readily parallelizable, because it uses a recurrence relation to generate the encryption mask. The schemes of [2, 33] can benefit from parallel processing. However, both consistently require more point additions compared to the proposed scheme, because they require on EC point multiplication per pixel [33], or group of pixels [2].

Given a certain parallel computer architecture, the speedup of a program is defined as the total time required to finish the equivalent sequential program divided by the time required to finish the parallel program. Assuming that the initialization overhead of the proposed scheme is trivial compared to the actual encryption load, the speedup of the proposed scheme is approximately equal to the number of available parallel processors, m.

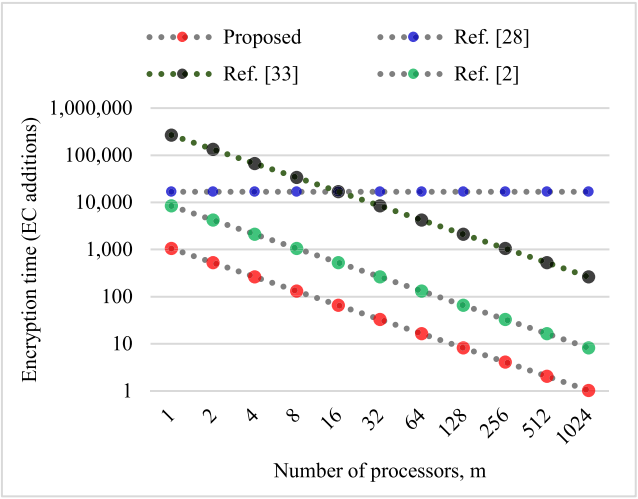

FIGURE 8. Comparison of theoretical computation time in terms of the number of EC point additions per parallel processor.

2) EXPERIMENTAL SPEED RESULTS

To demonstrate the practical benefits of the proposed parallelizable image encryption scheme, we implement it in Java and execute it on a PC with a quad-core Intel Core-i7 processor. Java facilitates parallel processing by providing logical execution units called threads. Each thread can then be executed independently by a physical processor. If more than one physical processor are available, multiple threads can be executed simultaneously in parallel. In our experiments, we vary the number of execution threads from 1 to 16 and observe the achieved speedup. Table 12 lists the encryption throughput and speedup as the number of threads increase. Figure 9 shows that the speedup resulting from parallel execution on the quad-core processor increases with the number of execution threads, achieving up to 3.93 speedup in encryption throughput with 16 parallel execution threads. Given that the computer has four independent processing cores, the achieved speedup is close to the expected theoretical speedup, $m = 4$.

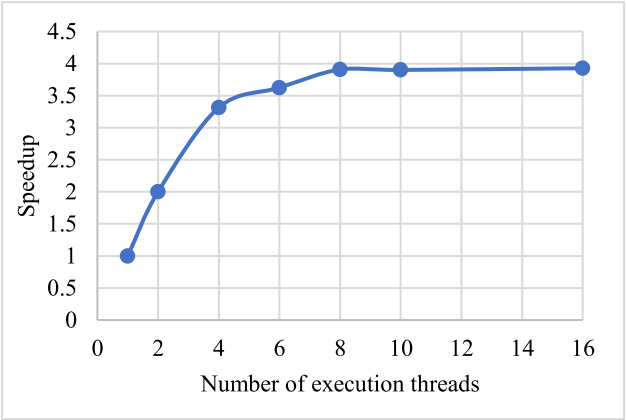

FIGURE 9. Image encryption speedup with parallel threads of execution.

TABLE 12. Experimental results of encryption time of a 1024 × 1024 image with varying number of threads and the corresponding speedup.

Threads	Encryption Time (s)	Throughput (MB/s)	Speedup
1	1.75	0.60	1.00
2	0.87	1.20	2.00
4	0.53	1.99	3.31
6	0.48	2.17	3.62
8	0.45	2.35	3.91
10	0.45	2.34	3.90
16	0.44	2.36	3.93

TABLE 13. Comparison between the proposed scheme and relevant parallel image encryption schemes.

Scheme	Hardware/software	Speedup	Efficiency
Proposed	Quad-core / Java JDK 15	3.93	98.3%
Ref. [12], 2020	Quad-core / C++	2.5	62.5%
Ref. [15], 2016	Quad-core / OpenMP, C++	3.64	91%
Ref. [25], 2017	Quad-core / OpenMP, C++	3.73	93.2%

TABLE 14. Experimental results of partial decryption time for a 1024 × 1024 image with varying percentage of decrypted area.

Decryption Area	Decryption time (ms)	Fraction of total encryption time
229 × 229 ≅ 5%	79.92	5.2%
324 × 324 ≅ 10%	154.40	10.1%
397 × 397 ≅ 15%	244.77	16.1%
458 × 458 ≅ 20%	308.44	20.2%
512 × 512 ≅ 25%	384.50	25.2%
561 × 561 ≅ 30%	459.86	30.2%
606 × 606 ≅ 35%	539.19	35.4%
648 × 648 ≅ 40%	613.52	40.2%
687 × 687 ≅ 45%	690.18	45.3%
724 × 724 ≅ 50%	771.77	50.6%
1024×1024 ≅ 100%	1524.76	100.0%

The efficiency of parallelization is given by the formula

$$efficiency = \frac{speed\ up}{number\ of\ processors} \times 100\% \qquad (12)$$

Table 13 emphasizes the superior parallelization efficiency of the proposed scheme in comparison with the efficiency of relevant parallel image encryption schemes using multicore CPUs.

3) EFFICIENCY OF PARTIAL IMAGE DECRYPTION

One of the advantages of the proposed scheme is the capability for partial decryption. To measure the efficiency of partial decryption we decrypt a small square of pixels with an arbitrary percentage of the area of the cipher image and compare the decryption time to the time required to decrypt the entire image. We vary the decrypted area between 5% and 50% of the image and record the decryption time as a fraction of the total decryption time of the cipher image. Results shown in Table 14 indicate that the decryption time is directly proportional to the area that needs to be decrypted. Therefore, the proposed scheme saves time and work when only a specific part of the cipher image needs to be decrypted.

V. CONCLUSION

An important limitation of most chaotic maps is their inherent sequential computation, which prevents them from taking advantage from advances in parallel processing capabilities now available on most computing platforms. In this paper, we proposed a highly parallel image encryption scheme, which utilizes available parallel processing resources to reduce encryption and decryption times. The proposed scheme avoids the sequential dependency of traditional chaotic systems by using an EC-based chaotic system, in which recurrent point additions are replaced by parallel additions. The proposed scheme achieved a speedup of 3.93 on a quad-core microprocessor. The achieved speedup is equivalent to 98.3% parallelism efficiency, which exceeds existing parallel image encryption schemes. In addition to enabling parallel processing to increase encryption and decryption throughput, the proposed scheme enables partial image decryption avoiding the unnecessary cost of full decryption. The proposed scheme was shown to resist common cryptanalysis attacks including chosen plaintext attacks. Theoretically, the proposed scheme can efficiently use massively parallel computing architectures with hundreds or thousands of cores. The practical considerations of programming such a massively parallel implementation is left for future work.

REFERENCES

[1] R. Matthews, "On the derivation of a 'chaotic' encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, Jan. 1989.
[2] D. S. Laiphrakpam and M. S. Khumanthem, "A robust image encryption scheme based on chaotic system and elliptic curve over finite field," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8629–8652, Apr. 2018.
[3] X. Wang, N. Guan, H. Zhao, S. Wang, and Y. Zhang, "A new image encryption scheme based on coupling map lattices with mixed multi-chaos," *Sci. Rep.*, vol. 10, no. 1, p. 9784, Jun. 2020.
[4] H. Zhu, Y. Zhao, and Y. Song, "2D logistic-modulated-sine-coupling-logistic chaotic map for image encryption," *IEEE Access*, vol. 7, pp. 14081–14098, 2019.
[5] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," *Inf. Sci.*, vol. 480, pp. 403–419, Apr. 2019.
[6] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan, and A. Zengin, "Secure image encryption algorithm design using a novel chaos based S-box," *Chaos, Solitons Fractals*, vol. 95, pp. 92–101, Feb. 2017.
[7] X. Zhang, Y. Mao, and Z. Zhao, "An efficient chaotic image encryption based on alternate circular S-boxes," *Nonlinear Dyn.*, vol. 78, no. 1, pp. 359–369, Oct. 2014.

[8] X. Wang, Ü. Çavuşoğlu, S. Kacar, A. Akgul, V.-T. Pham, S. Jafari, F. Alsaadi, and X. Nguyen, "S-box based image encryption application using a chaotic system without equilibrium," *Appl. Sci.*, vol. 9, no. 4, p. 781, Feb. 2019.

[9] X.-P. Zhang, R. Guo, H.-W. Chen, Z.-M. Zhao, and J.-Y. Wang, "Efficient image encryption scheme with synchronous substitution and diffusion based on double S-boxes," *Chin. Phys. B*, vol. 27, no. 8, Aug. 2018, Art. no. 080701.

[10] S. Ibrahim, H. Alhumyani, M. Masud, S. S. Alshamrani, O. Cheikhrouhou, G. Muhammad, M. S. Hossain, and A. M. Abbas, "Framework for efficient medical image encryption using dynamic S-boxes and chaotic maps," *IEEE Access*, vol. 8, pp. 160433–160449, 2020.

[11] S. Ibrahim and A. Alharbi, "Efficient image encryption scheme using Henon map, dynamic S-boxes and elliptic curve cryptography," *IEEE Access*, vol. 8, pp. 194289–194302, 2020.

[12] W. Song, Y. Zheng, C. Fu, and P. Shan, "A novel batch image encryption algorithm using parallel computing," *Inf. Sci.*, vol. 518, pp. 211–224, May 2020.

[13] A. Elrefaey, A. Sarhan, and N. M. El-Shennawy, "Parallel approaches to improve the speed of chaotic-maps-based encryption using GPU," *J. Real-Time Image Process.*, 2021.

[14] J. Liu, H. Z. T. , D. Song, G. Sun, W. C. Bi, and M. K. Buza, "A parallel encryption algorithm of the logistic map for multicore with OpenMP," presented at the Ifost, Ulaanbaatar, Mongolia, 2013.

[15] D. Burak, "Parallelization of image encryption algorithm based on chaotic neural networks," in *Artificial Intelligence and Soft Computing—ICAISC* (Lecture Notes in Computer Science), L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds. Cham, Switzerland: Springer, 2016.

[16] M. J. Rostami, A. Shahba, S. Saryazdi, and H. Nezamabadi-Pour, "A novel parallel image encryption with chaotic windows based on logistic map," *Comput. Electr. Eng.*, vol. 62, pp. 384–400, Aug. 2017.

[17] X. Wang, L. Feng, and H. Zhao, "Fast image encryption algorithm based on parallel computing system," *Inf. Sci.*, vol. 486, pp. 340–358, Jun. 2019.

[18] L. You, E. Yang, and G. Wang, "A novel parallel image encryption algorithm based on hybrid chaotic maps with OpenCL implementation," *Soft Comput.*, vol. 24, no. 16, pp. 12413–12427, Aug. 2020.

[19] Q. Zhou, K.-W. Wong, X. Liao, T. Xiang, and Y. Hu, "Parallel image encryption algorithm based on discretized chaotic map," *Chaos, Solitons Fractals*, vol. 38, no. 4, pp. 1081–1092, Nov. 2008.

[20] Y. Luo, R. Zhou, J. Liu, Y. Cao, and X. Ding, "A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map," *Nonlinear Dyn.*, vol. 93, no. 3, pp. 1165–1181, Aug. 2018.

[21] Ü. Çavuşoğlu and S. Kaçar, "A novel parallel image encryption algorithm based on chaos," *Cluster Comput.*, vol. 22, no. 4, pp. 1211–1223, Dec. 2019.

[22] C. Li, G. Luo, and C. Li, "A parallel image encryption algorithm based on chaotic duffing oscillators," *Multimedia Tools Appl.*, vol. 77, no. 15, pp. 19193–19208, Aug. 2018.

[23] X. Wang and H. Zhao, "Fast image encryption algorithm based on parallel permutation-and-diffusion strategy," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 19005–19024, Jul. 2020.

[24] S. Mozaffari, "Parallel image encryption with bitplane decomposition and genetic algorithm," *Multimedia Tools Appl.*, vol. 77, no. 19, pp. 25799–25819, Oct. 2018.

[25] D. Burak, "Parallelization of image encryption algorithm based on game of life and chaotic system," in *Artificial Intelligence and Soft Computing—ICAISC* (Lecture Notes in Computer Science), L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds. Cham, Switzerland: Springer, 2017.

[26] P. Ping, X. Zhang, X. Yang, Y. Mao, and Z. Gao, "Parallel image encryption technology based on cellular automaton," presented at the IEEE 6th Int. Conf. Big Data Comput. Service Appl. (BigDataService), Oxford, U.K., 2020.

[27] X. Wang and L. Liu, "Cryptanalysis of a parallel sub-image encryption method with high-dimensional chaos," *Nonlinear Dyn.*, vol. 73, nos. 1–2, pp. 795–800, Jul. 2013.

[28] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Process.*, vol. 141, pp. 217–227, Dec. 2017.

[29] U. Hayat and N. A. Azam, "A novel image encryption scheme based on an elliptic curve," *Signal Process.*, vol. 155, pp. 391–402, Feb. 2019.

[30] Z. E. Dawahdeh, S. N. Yaakob, and R. R. Bin Othman, "A new image encryption technique combining elliptic curve cryptosystem with hill cipher," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 30, no. 3, pp. 349–355, Jul. 2018.

[31] X. Zhang and X. Wang, "Digital image encryption algorithm based on elliptic curve public cryptosystem," *IEEE Access*, vol. 6, pp. 70025–70034, 2018.

[32] Y. Luo, X. Ouyang, J. Liu, and L. Cao, "An image encryption method based on elliptic curve Elgamal encryption and chaotic systems," *IEEE Access*, vol. 7, pp. 38507–38522, 2019.

[33] J. Wu, X. Liao, and B. Yang, "Color image encryption based on chaotic systems and elliptic curve ElGamal scheme," *Signal Process.*, vol. 141, pp. 109–124, Dec. 2017.

[34] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer-Verlag, 2004.

[35] P. L. Montgomery, "Speeding the Pollard and elliptic curve methods of factorization," *Math. Comput.*, vol. 48, no. 177, p. 243, Jan. 1987.

[36] B. S. Kaliski, "A pseudo-random bit generator based on elliptic logarithms," in *Advances in Cryptology—CRYPTO*. Berlin, Germany: Springer, 1987, pp. 84–103.

[37] S. Hallgren, *Linear Congruential Generators Over Elliptic Curves*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 2001.

[38] L.-P. Lee and K.-W. Wong, "A random number generator based on elliptic curve operations," *Comput. Math. Appl.*, vol. 47, nos. 2–3, pp. 217–226, Jan. 2004.

[39] T. Lange and I. E. Shparlinski, "Certain exponential sums and random walks on elliptic curves," *Can. J. Math.*, vol. 57, no. 2, pp. 338–350, Apr. 2005.

[40] E. B. Barker, W. C. Barker, W. E. Burr, W. T. Polk, and M. E. Smid, *Recommendation for Key Management, Part 1: General (Revised)*, Standard SP 800-57, National Institute of Standards & Technology, 2007.

[41] B. Schoenmakers and A. Sidorenko, "Cryptanalysis of the dual elliptic curve pseudorandom generator," *IACR Cryptol. ePrint Arch.*, vol. 2006, p. 190, May 2006.

[42] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: A year in review," *J. Inf. Secur. Appl.*, vol. 48, Oct. 2019, Art. no. 102361.

[43] S. Ibrahim and A. M. Abbas, "Efficient key-dependent dynamic S-boxes based on permutated elliptic curves," *Inf. Sci.*, vol. 558, pp. 246–264, May 2021.

[44] E. E. Mahassni and I. Shparlinski, "On the uniformity of distribution of congruential generators over elliptic curves," in *Sequences and Their Applications*. London, U.K.: Springer, 2002, pp. 257–264.

[45] A. Belazi, M. Talha, S. Kharbech, and W. Xiang, "Novel medical image encryption scheme based on chaos and DNA encoding," *IEEE Access*, vol. 7, pp. 36667–36681, 2019.

[46] L. E. Bassham, III *et al.*, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Standard SP 800-22 Rev. 1a, National Institute of Standards & Technology, 2010.

[47] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, and M. F. Abu-ElYazeed, "A novel image encryption system merging fractional-order edge detection and generalized chaotic maps," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107280.

[48] X. Chai, J. Zhang, Z. Gan, and Y. Zhang, "Medical image encryption algorithm based on Latin square and memristive chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 24, pp. 35419–35453, Dec. 2019.

[49] M. Asgari-Chenaghlu, M.-A. Balafar, and M.-R. Feizi-Derakhshi, "A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation," *Signal Process.*, vol. 157, pp. 1–13, Apr. 2019.

[50] M. Alawida, A. Samsudin, J. S. Teh, and R. S. Alkhawaldeh, "A new hybrid digital chaotic system with applications in image encryption," *Signal Process.*, vol. 160, pp. 45–58, Jul. 2019.

[51] J. A. P. Artiles, D. P. B. Chaves, and C. Pimentel, "Image encryption using block cipher and chaotic sequences," *Signal Process., Image Commun.*, vol. 79, pp. 24–31, Nov. 2019.

[52] D. S. Laiphrakpam and M. S. Khumanthem, "Medical image encryption based on improved ElGamal encryption technique," *Optik*, vol. 147, pp. 88–102, Oct. 2017.

[53] A. Belazi, A. A. Abd El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Process.*, vol. 128, pp. 155–170, Nov. 2016.

[54] J. B. Lima, F. Madeiro, and F. J. R. Sales, "Encryption of medical images based on the cosine number transform," *Signal Process., Image Commun.*, vol. 35, pp. 1–8, Jul. 2015.
[55] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "Generalized majority logic criterion to analyze the statistical strength of S-boxes," *Zeitschrift für Naturforschung A*, vol. 67, no. 5, pp. 282–288, May 2012.
[56] Y. Chen, C. Tang, and R. Ye, "Cryptanalysis and improvement of medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Process.*, vol. 167, Feb. 2020, Art. no. 107286.
[57] Z. M. Z. Muhammad and F. Özkaynak, "An image encryption algorithm based on chaotic selection of robust cryptographic primitives," *IEEE Access*, vol. 8, pp. 56581–56589, 2020.
[58] Z. Hua, B. Xu, F. Jin, and H. Huang, "Image encryption using Josephus problem and filtering diffusion," *IEEE Access*, vol. 7, pp. 8660–8674, 2019.
[59] A. Ullah, S. S. Jamal, and T. Shah, "A novel scheme for image encryption using substitution box and chaotic system," *Nonlinear Dyn.*, vol. 91, no. 1, pp. 359–370, Jan. 2018.

ALAA M. ABBAS received the Ph.D. degree from Menofia University, Egypt, in 2008. He is currently an Associate Professor with the Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menofia University. He is also an Assistant Professor with the Electrical Engineering Department, College of Engineering, Taif University, Saudi Arabia. His research interests include image processing, watermarking, image encryption, and cryptography.

AYMAN A. ALHARBI received the B.Sc. degree from Umm Al-Qura University, Saudi Arabia, in 2006, and the M.Sc. and Ph.D. degrees in computer science and engineering from the University of Connecticut, USA, in 2012 and 2015, respectively.

He was the Chairman of the Computer Engineering Department. He also worked as the Vice-Principal of the Department of Investment, Umm Al-Qura University. Since 2010, he has been a member with the UConn's Underwater Sensor Networks Laboratory. He is currently an Assistant Professor with Umm Al-Qura University.

SALEH IBRAHIM received the B.Sc. and M.Sc. degrees in computer engineering from Cairo University, Egypt, in 2000 and 2004, respectively, and the Ph.D. degree in computer science and engineering from the University of Connecticut, USA, in 2010.

He is currently an Assistant Professor with the Electrical Engineering Department, Taif University, Saudi Arabia. He has been an Assistant Professor with the Computer Engineering Department, Cairo University, since 2011. He has published several research papers in high-impact journals and international conferences. His current research interests include information security and computer networks.