

Received October 5, 2019, accepted October 24, 2019, date of publication October 29, 2019, date of current version November 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950007

Chaotic Encryption Algorithm With Key Controlled Neural Networks for Intelligent Transportation Systems

GRAHAM R. W. THOMS¹, (Student Member, IEEE), RADU MURESAN¹, (Member, IEEE), AND ARAFAT AL-DWEIK^{2,3}, (Senior Member, IEEE)

¹School of Engineering, University of Guelph, Guelph, ON N1G 2W1, Canada

²Center for Cyber Physical Systems (C2PS), Khalifa University, Abu Dhabi 127788, United Arab Emirates

³Department of Electrical and Computer Engineering, Western University, London, ON N6A 3K7, Canada

Corresponding author: Arafat Al-Dweik (arafat.dweik@ku.ac.ae)

This work was supported by the Ministry of Transportation of Ontario (MTO), Highway Infrastructure Innovation Funding Program (HIIFP) under Grant 051938. The work of A. Al-Dweik was supported by the Khalifa University Center for Cyber Physical Systems (C2PS).

ABSTRACT The security of sensitive information is vital in many aspects of multimedia applications such as Intelligent Transportation Systems (ITSs), where traffic data collection, analysis and manipulations is essential. In ITS, the images captured by roadside units form the basis of many traffic rerouting and management techniques, and hence, we should take all precautions necessary to deter unwanted traffic actions caused by malicious adversaries. Moreover, the collected traffic images might reveal critical private information. Consequently, this paper presents a new image encryption algorithm, denoted as ChaosNet, using chaotic key controlled neural networks for integration with the roadside units of ITSs. The encryption algorithm is based on the Lorenz chaotic system and the novel key controlled finite field neural network. The obtained cryptanalysis show that the proposed encryption scheme has substantial mixing properties, and thus cryptographic strength with up to 5% increase in information entropy compared to other algorithms. Moreover, it offers consistent resistance to common attacks demonstrated by nearly ideal number of changing pixel rate (NPCR), unified averaged changed intensity (UACI), pixel correlation coefficient values, and robustness to cropped attacks. Furthermore, it has less than 0.002% difference in the NPCR and 0.3% in the UACI metrics for different test images.

INDEX TERMS Neural network encryption, image encryption, cryptography, finite fields, chaotic systems, intelligent transportation systems, smart city, IoT, Internet of Things.

I. INTRODUCTION

Advanced intelligent transportation systems (ITSs) are one of the main driving technologies of smart city development, becoming a pillar of city infrastructure as population and autonomous vehicle developments continue to grow [1]. The objectives of an ITS can vary widely, and may include traffic detection, control and analysis. Public data collection, traffic management, localized alerts and real-time traffic management are major services that are being developed and have mission critical functions. Therefore, security and safety aspects of an ITS become a main concern. A safe ITS must prevent malicious attackers from altering sensitive traffic data that could produce unwanted traffic actions, and must keep the collected information secured at all stages.

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Fadda¹.

In this sequel, this paper presents the development of an image encryption algorithm, ChaosNet, utilizing chaotic systems and key-controlled neural networks for use in the Scalable Enhanced Roadside Unit (SERSU) [2] and other ITS applications.

For several years, chaotic maps have been an attractive basis for cryptographic applications, due to the hyper-sensitivity to initial conditions and input parameters, producing pseudorandom and unpredictable behavior [3]. The generation of these new chaos-based encryption schemes mainly focus on an image as the input, since many chaotic maps provide thorough topologically mixing properties which are well suited for the two-dimensional nature of images [4]–[20].

Chaotic encryption schemes can generally be split into three main categories. Firstly they can employ chaos as a mean of performing complex permutations of coordinates

with repeated iterations. Early adopters of chaos theory in encryption schemes utilized this method of chaotic coordinate transformation to topologically mix the image as much as possible, as in the case of the baker map [21]. The Baker map exploits a ‘kneeling’ and ‘folding’ of two dimensional data iterated several times. This map is discretized for digital images and generalized to incorporate a secret key for the encryption process. Furthermore the map is extended into three dimensions to increase mixing of grey levels to produce an adequate encryption scheme.

Secondly, chaotic systems can employ complex value substitutions within the plaintext, needed for the confusion component of the algorithm. In [15], the authors utilize the Colpitts system with chaos inducing parameters and the Duffing chaotic system. With these systems they produce a pseudo-random two dimensional chaotic map with the same dimensions as the plainimage. Each plaintext pixel is then mapped 1-to-1 to the chaotic substitution lattice and iterated several times.

Recently, chaotic maps are designed by combining the two previous methods. That is, chaotic maps are exercised as complex substitutions and permutations in a combined encryption algorithm to achieve adequate diffusion and confusion properties for optimal cipher security. In [22], the authors utilize multiple logistic maps as their anchor of chaos. The first logistic map is utilized as a pixel permuting scheme, while the other logistic map handles the pixel value transformation, or the substitution scheme. In [23], there are distinct sections for chaotic permutation and substitution in the cipher. The authors utilize the Arnold cat map as a complex iterative permutation element with the use of two secret keys. The data is then fed to multiple cascaded discrete duffing equations with associated keys for pixel value substitution. Furthermore, chaotic coordinate and pixel transformation can also be infused with other complex functions. Such as in [3], the chaotic tent-map is modified to incorporate the rectangular transform. This is used as a plaintext permutation section, while the chaotic tent map itself generates a sequence for use in the pixel value substitution section. In [24], the authors combine image encryption with an image compression scheme based on compressive sensing with the Walsh-Hadamard transform along with two chaotic maps each created from a combination of the Logistic map, Tent map, and Sine map. The compressed image is then permuted according to a pseudorandom sequence and undergoes complex substitution based on DNA sequence operations to produce the final encrypted image. In [25], the authors introduce a phase-truncated short-time fractional Fourier Transform within an encryption unit along with wave-based image permutation and complex substitution through Chen’s hyper-chaotic system.

This paper develops an image encryption algorithm based on the Lorenz chaotic system and chained finite field transformation layers to form a neural-network-like structure, which in conjunction with a novel full-image permutation scheme will produce an encryption algorithm that makes

cipherimages unintelligible and highly secure. The algorithm, analyzed in terms of cryptographic strength through various metrics and tests, and the obtained results show to be a suitable image encryption scheme for an ITS.

The remainder of this paper is organized as follows. Section II outlines necessary background components involved in ChaosNet, including the Lorenz system, Galois Field 2^8 , the neural network Hill cipher. In Section III, the details of the ChaosNet image encryption scheme are outlined. In Section IV, the security of ChaosNet is analyzed through various relevant tests and compared to previous image encryption algorithms, and finally the findings are summarized and concluded in Section VI.

II. CHAOSNET COMPONENTS

A. CHAOTIC SYSTEMS

Chaotic systems exhibit emergence over scale, however, they do so in an unpredictable yet deterministic manner. Where common patterns may emerge, exact positions are highly unpredictable which are caused by hypersensitive dependence to the systems initial conditions. Chaotic systems are suitable for image encryption due to high initial condition sensitivity, randomness, unpredictability and are topologically mixing [26].

Chaos can also be measured quantitatively, through means of the Lyapunov characteristic exponent(s) (LCE) of a given dynamic system, which describe the trajectory evolution of a dynamic (discrete) system,

$$\gamma(x_0) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \quad (1)$$

where $f(x_i)$ gives the subsequent point x_{i+1} , producing the Lyapunov exponent for the dynamic variable x . A spectrum of Lyapunov exponents is produced depending on the size of the phase space, i.e. a system with three dynamic equations will yield three Lyapunov exponents. The most important indicator for a system to be chaotic is to have at least one positive exponent within the set of resulting Lyapunov exponents, which indicates trajectory divergence and quantifies dependence on initial conditions by showing the rate at which two close points diverge over time [27].

B. LORENZ SYSTEM

The Lorenz strange (chaotic) attractor is a set of first order differential equations that describe a three dimensional point,

$$\begin{aligned} \frac{dx}{dt} &= \alpha(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \quad (2)$$

using real parameters α , ρ , and β . The Lorenz system is extremely sensitive to initial coordinate positions, serving unpredictable yet bounded behaviour. The Lyapunov exponents of the Lorenz system for $\alpha = 10$, $\beta = \frac{8}{3}$ and

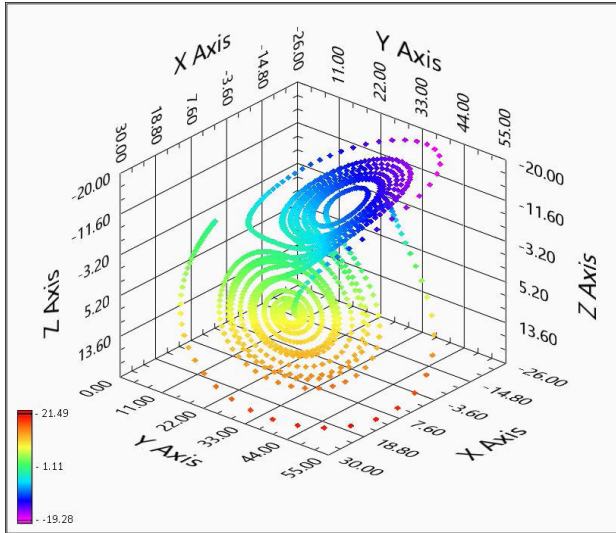


FIGURE 1. Plot of the Lorenz attractor given that $\alpha = 10$, $\beta = \frac{8}{3}$, $\rho = 28$.

$\rho = 28$ is shown in Fig. 1 using the parameters $\{0.9056, 0, -14.5723\}$ [28].

C. FINITE FIELD

A finite field is a finite set with definitions for addition, subtraction, multiplication and division. The number of elements in a field defines the field's order, which in this paper must be prime p to the n^{th} degree, known as a Galois Field $GF(p^n)$ [29]. The finite fields $GF(p^n)$ and its associated operations provide the necessary methods for combining two integers in the range $0, 1, \dots, p^n - 1$, to form a unique irreducible integer in that field, while also being able to calculate its inverse. Elements in $GF(p^n)$ represent the powers of a polynomial, as polynomial arithmetic defines the operations within $GF(p^n)$.

The proposed chaotic weight matrix will be generated in $GF(2^8)$, thus each element of the weight matrix will represent a polynomial within $GF(2^8)$. This choice is convenient because the grayscale images used for encryption use 8-bit gray levels. By representing the elements of the weight matrix and input in $GF(2^8)$, the weight matrix will be irreducible with the irreducible polynomial that defines the field. These polynomials are represented as 8-bit integers such that each bit position corresponds to a term within the polynomial. For example the decimal number 157 represents the polynomial in $GF(2^8)$ as,

$$156_{10} = 9C_{16} = 10011101_2 = x^7 + x^4 + x^3 + x^2 + 1.$$

The normal addition/subtraction and multiplication/division operations of real numbers follow the field rules of $GF(2^8)$ [29]. Addition and subtraction of two elements result in the same operation, a bitwise XOR of those two elements. For example the addition or subtraction of the elements $3F_{16}$ and $A5_{16}$ in $GF(2^8)$ is,

$$3F_{16} \oplus A5_{16} = 9A_{16}.$$

Multiplication of two elements becomes a polynomial multiplication of those two elements modulo an irreducible polynomial in $GF(2^8)$. Division is the same as multiplication however the first factor is multiplied by the multiplicative inverse of the second factor modulo the irreducible polynomial. The irreducible polynomial in $GF(2^8)$ used in this algorithm is the one used by the Advanced Encryption Standard (AES) [30],

$$11B_{16} = 283_{10} = 100011011_2 = x^8 + x^4 + x^3 + x + 1. \quad (3)$$

For example, the multiplication, denoted as \cdot , of $3F_{16}$ and $A5_{16}$ in $GF(2^8)$ is shown below,

$$\begin{aligned} 3F_{16} \cdot A5_{16} &= 00111111_2 \cdot 10100101_2 \\ &= (x^5 + x^4 + x^3 + x^2 + x + 1) \\ &\quad \cdot (x^7 + x^5 + x^2 + 1) \\ &= x^{12} + x^{11} + 2x^{10} + 2x^9 + 2x^8 \\ &\quad + 3x^7 + 2x^6 + 3x^5 + 2x^4 + 2x^3 + 2x^2 + x + 1 \end{aligned} \quad (4)$$

since addition in $GF(2^8)$ is an XOR, the terms with even coefficients can be eliminated,

$$\begin{aligned} 3F_{16} \cdot A5_{16} &= x^{12} + x^{11} + x^7 + x^5 + x + 1 \\ &= 1100010100011_2 \end{aligned} \quad (5)$$

this subsequently must be modded by the irreducible polynomial (3), which can be done by long division with XOR in place of subtraction,

$$\begin{aligned} &= 1100010100011 \quad \text{mod } 10011101 \\ &\oplus 100011011 \\ &\quad 100100010011 \\ &\oplus 100011011 \\ &\quad 111001011 \\ &\oplus 100011011 \\ &\quad 11010000 = D0_{16}. \end{aligned}$$

Therefore in $GF(2^8)$, $3F_{16} \cdot A5_{16} = D0_{16}$.

D. NEURAL NETWORK HILL CIPHER

The complex substitution sub-algorithm incorporates chained hill ciphers that resemble a fully connected neural network structure given by the equation,

$$y_i = \sum_{j=1}^N w_{ij}x_j \quad \forall i = 0, 1, \dots, M. \quad (6)$$

However the weights and inputs, w_{ij} and x_{ij} , will be treated as elements of $GF(2^8)$ as described in Section II-C. Thus, addition and multiplication operations shown in (6) are in $GF(2^8)$. The matrix form of (6) is given by

$$y = Wx$$

$$y = Wx$$

$$\begin{bmatrix} 85 \\ 8c \\ 60 \\ 27 \\ e3 \\ b2 \\ 46 \\ cd \\ b1 \\ c5 \\ 8f \\ d4 \\ 2e \\ 04 \\ 48 \\ 22 \end{bmatrix}_{16} = \begin{bmatrix} b1 & 83 & 06 & 84 & 0c & 02 & e9 & 12 & 13 & 4c & a3 & 30 & 3a & ed & a6 & 3c \\ cf & cd & f1 & 5f & cd & 39 & 80 & 35 & 8a & f6 & 3b & 05 & 41 & 0d & 95 & ca \\ a3 & 9d & 78 & 07 & 50 & 62 & 27 & 44 & 27 & af & ba & a3 & 3c & 37 & 5d & 84 \\ e8 & b9 & 2b & e0 & e7 & e6 & 8a & 9f & d4 & be & 02 & ad & 90 & cf & ed & 83 \\ 93 & a1 & 3a & 81 & e4 & 44 & 7f & 5d & a1 & f7 & 85 & 5b & 37 & d9 & fe & e7 \\ 0f & 62 & 49 & 0f & 25 & 7c & 49 & f8 & 1f & 0e & 2f & 7e & 6c & 0d & 89 & 15 \\ ef & ab & 4d & 7a & 35 & 7c & f8 & 70 & e7 & d6 & a2 & 8b & 07 & c8 & 0e & 71 \\ cf & b9 & 04 & 15 & 45 & 68 & c7 & 45 & 8b & 35 & e0 & 91 & 01 & 6c & e1 & da \\ 4f & 9e & 65 & 91 & 94 & 2f & a6 & 06 & d9 & 3c & dd & cf & a1 & 54 & 76 & 46 \\ 03 & be & 3c & 64 & e7 & 97 & ac & 3b & a0 & 8e & ef & 2d & 5b & 83 & 2e & c6 \\ de & d8 & cd & b5 & 06 & 33 & c9 & c2 & 3c & a7 & 08 & 16 & fb & 4f & 12 & 83 \\ e4 & 47 & af & 57 & a8 & 0f & ea & 13 & 0d & 8c & 48 & 60 & b9 & 88 & 31 & 6e \\ 87 & 82 & a2 & 58 & de & 07 & 80 & bd & 35 & 0f & 6b & ed & 80 & 4f & ca & b8 \\ 91 & 4d & 71 & 50 & fb & db & 70 & 31 & a8 & 0c & 8b & 89 & e0 & 0a & 40 & 92 \\ 2f & 2e & 12 & fd & 65 & 77 & 5e & 57 & eb & 0e & d8 & c5 & 36 & 06 & 58 & 87 \\ f1 & f3 & 26 & fa & 66 & 03 & f2 & ec & db & 6d & 15 & c5 & 23 & 02 & 13 & e8 \end{bmatrix}_{16} \begin{bmatrix} 1d \\ 8f \\ ed \\ 2d \\ bf \\ cb \\ b0 \\ bf \\ 9f \\ fa \\ fe \\ 7b \\ 1c \\ fa \\ bd \\ ef \end{bmatrix}_{16}$$

FIGURE 2. An example of a weight matrix in ChaosNet with a 128-bit input x , and output y through matrix multiplication by W in $GF(2^8)$.

where

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \dots & w_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

$$w_{ij}, x_i \in \{0, 1, 2, \dots, 255\}, \quad (7)$$

which can be cascaded to act as individual layers of an L -layer network within ChaosNet,

$$y_L = W_L(W_{L-1}(W_{L-2}(\dots(W_0(x))))$$

with an example of a single layer weight matrix shown in Fig. 2.

III. CHAOSNET ALGORITHM

The proposed ChaosNet algorithm shown in Fig. 3 introduces the chained Hill cipher matrices that form a neural-network-like structure used as a complex substitution sub-algorithm, combined with a novel pixel permutation sub-algorithm. The elements of the weight matrices reside in a finite field, namely $GF(2^8)$, which are constructed from the secret key and chaotic sequence generator, thus the weight matrix operations will be in $GF(2^8)$. Both sub-algorithms are iterated for the desired number of rounds across equally sized blocks of the image to form the cipher image.

A. ENCRYPTION ALGORITHM

- 1) Choose a secret key with values of the set $\{d_\alpha, d_\rho, d_\beta, h_{key}\}$ with,

$$d_\alpha, d_\rho, d_\beta \in \{0, 1, \dots, 2^{10} - 1\}$$

$$h_{key} \in \{0, 1, \dots, 2^{8N} - 1\},$$

where N is the number of bytes in an input block of the image. The values $d_\alpha, d_\rho, d_\beta$ give the decimal place numbers for the parameters α, ρ, β of the Lorenz system (2), respectively. The value h_{key} provides a

hash value with a length proportional to the block size length.

- 2) Iterate equation (2) using parameters,

$$\alpha = 10 + (d_\alpha \times 10^{-3})$$

$$\beta = \frac{8}{3} + (d_\beta \times 10^{-3})$$

$$\rho = 28 + (d_\rho \times 10^{-3}),$$

for $S + N^2$ times with $S \in \{1000, \dots, \infty\}$. At each iteration i between $S + 1$ and N^2 , take the coordinate outputs $(x_i, y_i, z_i) \in \mathbb{R}$ of (2) and produce a chaotic integer sequence W_f ,

$$W_f = [w_1, w_2, \dots, w_i, \dots, w_{N^2-1}, w_{N^2}]$$

where each index of W_f , namely w_i , is computed with (2) such that,

$$w_i = (10^5 \times \sum x_{s+i}, y_{s+i}, z_{s+i}) \pmod{2^8}$$

$$\forall i = 1, 2, \dots, N^2$$

Thus W_f is an array of length N^2 containing chaotically generated integers in $GF(2^8)$.

- 3) Reshape W_f into a $N \times N$ square matrix W with left-to-right wrapping such that,

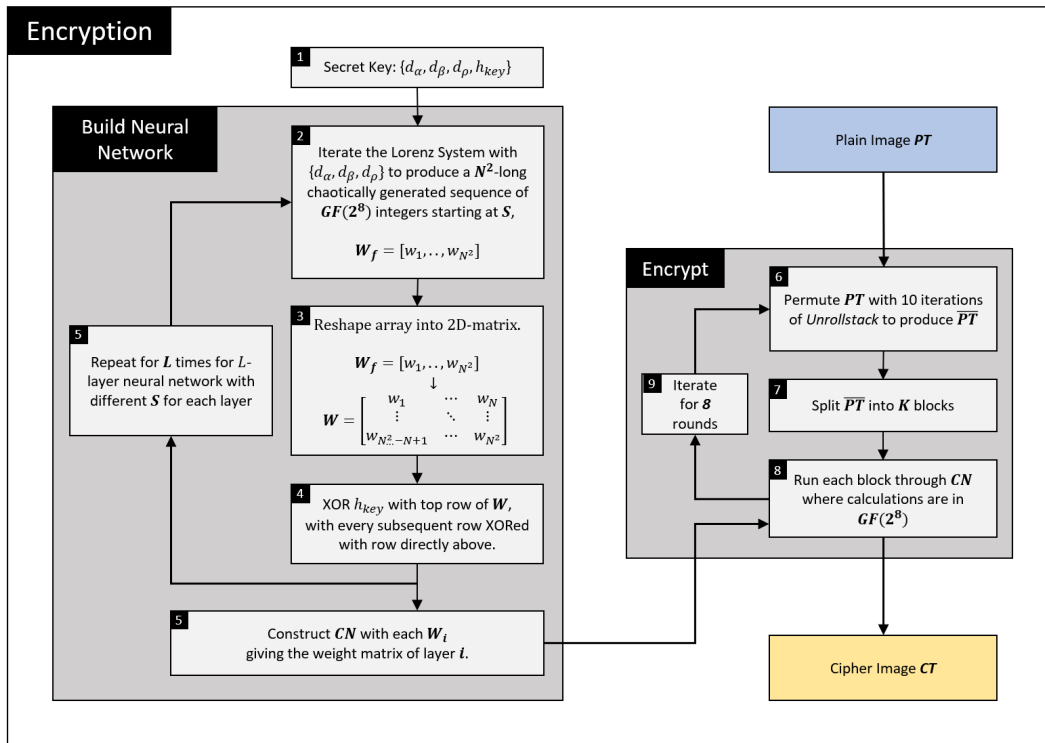
$$W_f = (w_1, w_2, \dots, w_i, \dots, w_{N^2-1}, w_{N^2})$$

$$\Downarrow$$

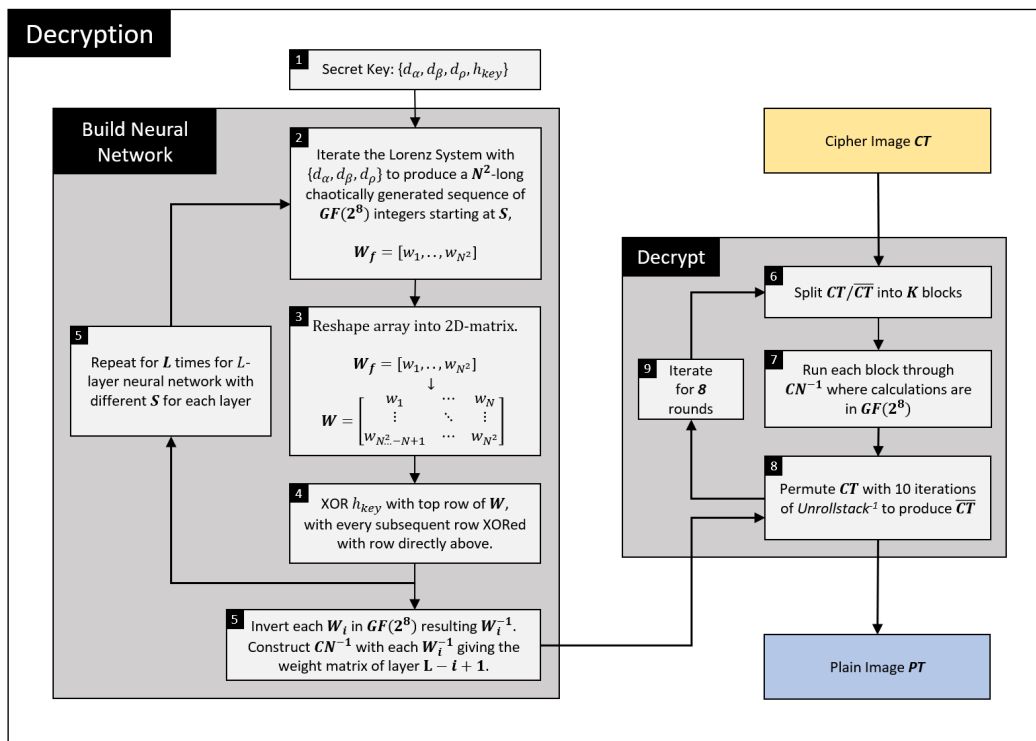
$$W = \begin{bmatrix} w_1 & w_2 & \dots & w_N \\ w_{N+1} & w_{N+2} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N^2-N+1} & w_{N^2-N+2} & \dots & w_{N^2} \end{bmatrix}.$$

- 4) Use h_{key} to XOR the top row of W and consecutively XOR each row with the row directly above such that,

$$W_{row(1)} = W_{row(1)} \oplus h_{key},$$



(a) ChaosNet Encryption



(b) ChaosNet Decryption

FIGURE 3. Overview of proposed ChaosNet (a) encryption and (b) decryption scheme.

with every subsequent row,

$$W_{row(i)} = W_{row(i)} \oplus W_{row(i-1)} \quad \forall i = 2, 3, \dots, N.$$

5) For a L -layer fully connected neural network, iterate steps 2-4 L times with each layer l using a unique S

such that,

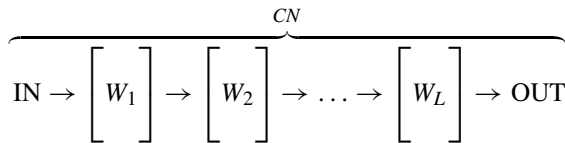
$$S_1 = 1000$$

$$S_{l+1} = S_l + N^2 \quad \forall l = 1, 2, \dots, L.$$

This will result in a set Φ of unique chaotically generated $N \times N$ weight matrices,

$$\Phi = \{W_1, W_2, \dots, W_i, \dots, W_{L-1}, W_L\}.$$

Assemble the L -layer neural network CN with Φ such that each W_i of Φ gives the weight matrices of layer i in the neural network. The matrix multiplication of IN with each subsequent weight matrix follows the multiplication described in Section II-C to generate the output OUT .



- 6) *Unrollstack*: Permute the 8-bit grayscale plain image PT by 'unrolling' its pixels starting from the outer right column working inwards, and then 'stacking' them from top to bottom. For example a single iteration of *Unrollstack* would transform a 3×3 image I into \bar{I} shown below,

$$I = \begin{bmatrix} \rightarrow 1 & \downarrow 2 & \downarrow 3 \\ \uparrow 4 & \downarrow 5 & \downarrow 6 \\ \uparrow 7 & \leftarrow 8 & \leftarrow 9 \end{bmatrix}$$

Unrollstack \Downarrow

$$\bar{I} = \begin{bmatrix} 3 \rightarrow & 6 \rightarrow & 9 \rightarrow \\ 8 \rightarrow & 7 \rightarrow & 4 \rightarrow \\ 1 \rightarrow & 2 \rightarrow & 5 \rightarrow \end{bmatrix}.$$

Permute PT in this fashion through 10 iterations of *Unrollstack* to produce \overline{PT} .

- 7) Split \overline{PT} into blocks $B = \{b_0, b_1, \dots, b_i, \dots, b_{K-2}, b_{K-1}\}$ for $K = \frac{\overline{PT}(\text{bytes})}{N}$, where each block b_i is N bytes.
- 8) Feed each block b_i through the CN from step 5. All elements of each weight matrix in CN are in $GF(2^8)$, thus perform each weight matrix multiplication in $GF(2^8)$. An example weight matrix multiplication in $GF(2^8)$ is shown in Fig. 2.
- 9) Repeat steps 6-8 for 8 rounds to produce the final cipherimage CT .

B. DECRYPTION ALGORITHM

- 1) Receive the secret keys, i.e. the set $\{d_\alpha, d_\rho, d_\beta, h_{key}\}$.
- 2) Iterate equation (2) using parameters,

$$\alpha = 10 + (d_\alpha \times 10^{-3})$$

$$\beta = \frac{8}{3} + (d_\beta \times 10^{-3})$$

$$\rho = 28 + (d_\rho \times 10^{-3}),$$

for $S + N^2$ times with $S \in \{1000, \dots, \infty\}$. At each iteration i between $S + 1$ and N^2 , take the coordinate

outputs $(x_i, y_i, z_i) \in \mathbb{R}$ of (2) and produce a chaotic integer sequence W_f ,

$$W_f = [w_1, w_2, \dots, w_i, \dots, w_{N^2-1}, w_{N^2}]$$

where each index of W_f , namely w_i , is computed with (2) such that,

$$w_i = (10^5 \times \sum x_{s+i}, y_{s+i}, z_{s+i}) \pmod{2^8}$$

$$\forall i = 1, 2, \dots, N^2$$

- 3) Reshape W_f into a $N \times N$ square matrix W with left-to-right wrapping such that,

$$W_f = (w_1, w_2, \dots, w_i, \dots, w_{N^2-1}, w_{N^2})$$

$$\Downarrow$$

$$W = \begin{bmatrix} w_1 & w_2 & \dots & w_N \\ w_{N+1} & w_{N+2} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N^2-N+1} & w_{N^2-N+2} & \dots & w_{N^2} \end{bmatrix}.$$

- 4) Use h_{key} to XOR the top row of W and consecutively XOR each row with the row directly above such that,

$$W_{row(1)} = W_{row(1)} \oplus h_{key},$$

with every subsequent row,

$$W_{row(i)} = W_{row(i)} \oplus W_{row(i-1)} \quad \forall i = 2, 3, \dots, N.$$

- 5) For a L -layer fully connected neural network, iterate steps 2-4 L times with each layer l using a unique S such that,

$$S_1 = 1000$$

$$S_{l+1} = S_l + N^2 \quad \forall l = 1, 2, \dots, L.$$

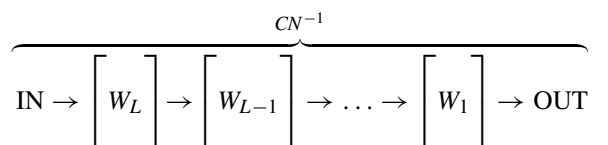
This will result in a set Φ of unique chaotically generated $N \times N$ weight matrices,

$$\Phi = \{W_1, W_2, \dots, W_i, \dots, W_{L-1}, W_L\}.$$

Produce a new set Φ^{-1} by taking the inverse of each W_i in $GF(2^8)$,

$$\Phi^{-1} = \{W_1^{-1}, W_2^{-1}, \dots, W_i^{-1}, \dots, W_{L-1}^{-1}, W_L^{-1}\}.$$

Assemble the L -layer neural network CN^{-1} with Φ^{-1} such that each W_i^{-1} of Φ^{-1} gives the weight matrices of layer $L - i + 1$ in the neural network. The matrix multiplication of IN with each subsequent weight matrix follows the multiplication described in Section II-C to generate the output OUT .



- 6) Split CT into blocks $B = \{b_0, b_1, \dots, b_i, \dots, b_{K-2}, b_{K-1}\}$ for $K = \frac{CT(\text{bytes})}{N}$, where each block b_i is N bytes.

- 7) Feed each block b_i through the CN^{-1} from step 5. All elements of each weight matrix in CN^{-1} are in $GF(2^8)$, thus perform each weight matrix multiplication in $GF(2^8)$.
- 8) *Unrollstack*⁻¹: Reverse the permutation transform of *Unrollstack* of the entire 8-bit grayscale image CT . For example a single iteration of *Unrollstack*⁻¹ would transform a 3×3 image I into \bar{I} shown below,

$$I = \begin{bmatrix} 3 \rightarrow & 6 \rightarrow & 9 \rightarrow \\ 8 \rightarrow & 7 \rightarrow & 4 \rightarrow \\ 1 \rightarrow & 2 \rightarrow & 5 \rightarrow \end{bmatrix}$$

Unrollstack⁻¹ ↓

$$\bar{I} = \begin{bmatrix} \rightarrow 1 & \downarrow 2 & \downarrow 3 \\ \uparrow 4 & \downarrow 5 & \downarrow 6 \\ \uparrow 7 & \leftarrow 8 & \leftarrow 9 \end{bmatrix}.$$

Permute CT in this fashion through 10 iterations of *Unrollstack*⁻¹ to produce \bar{CT} .

- 9) Repeat steps 6-8 for 8 rounds to produce the original plainimage PT .

IV. EXPERIMENTAL RESULTS

This Section provides the results of ChaosNet and its application in an ITS. It also provides the necessary image encryption tests for ChaosNet that prove its prototypical cryptographic strength. Such tests are typically used in such scenarios [31], [32]. Initial tests of encryption with ChaosNet are performed on highway traffic images taken from the SERSU [2] are shown in Table 2, where the distinct histograms of the plainimages become unintelligible cipherimages with flat histograms, increasing security.

A. INFORMATION ENTROPY

A term often used in image encryption and in cryptography is information entropy. Information entropy $H(x)$ is the average amount of information contained in a set of data, expressed as the average logarithm of a variable X with a probability distribution $P(X)$,

$$H(X) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (8)$$

where $P(X) = p(x_1), \dots, p(x_n)$. Information entropy (8) relates the probability of a variable to the amount of information in the variable, which can be thought of as a measure of randomness. The entropy of an image can be directly calculated from its histogram, since the histogram directly related to the probability distribution of the image. For a cryptic image scheme, extremely high information entropy is needed, bounded by codeword length. since high entropy will likely produce patternless data. In [33], it is stated that the uniform distribution produces the maximum entropy for a discrete random variable X . For example, the maximum information entropy of a variable with an 8 bit range (0, 1, 2, ..., 255) is 8 bits. A normal image will have an unbalanced distribution of pixel values,

TABLE 1. Entropy comparison of other chaos-based encryption algorithms.

Algorithm	Lena 256×256	Lena 512×512	Cameraman 256×256	Baboon 256×256
[11]	7.997	-	-	-
[34]	7.997	-	-	-
[35]	-	7.997	-	-
[36]	-	7.997	-	-
[15]	-	7.996	-	-
[22]	-	7.9993	-	-
[37]	-	-	7.571	-
[38]	-	-	-	7.9972
[39]	-	-	-	7.9994
[40]	-	-	-	7.9874
[41]	-	-	-	7.9952
[42]	-	-	-	7.9717
[43]	-	-	-	7.9968
ChaosNet	7.997	7.9993	7.997	7.9972

allowing for image encoding with fewer bits. This is due to the redundancy of pixel values, since adjacent pixels are likely to be similar. Shown in Table 1, ChaosNet is on par or better than other chaotic image encryption algorithms for the specified images.


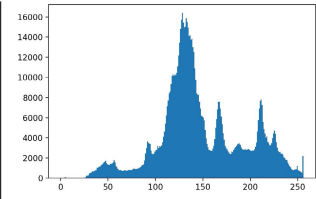
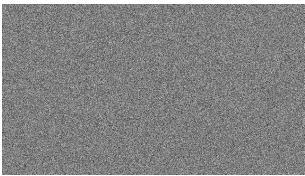
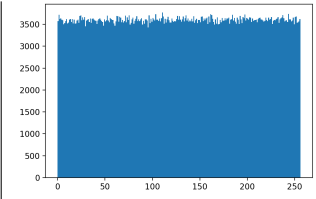

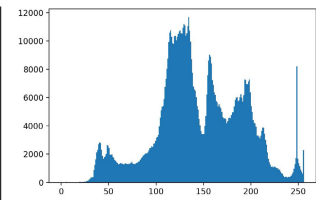
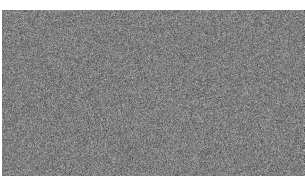
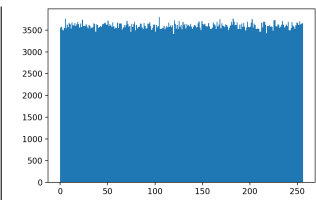
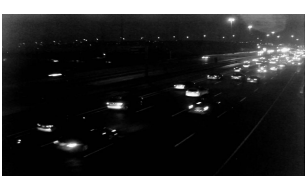
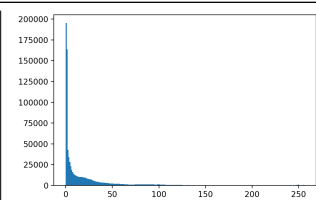

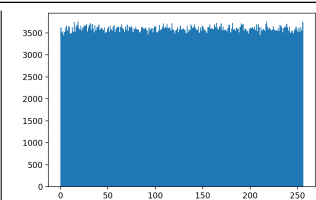

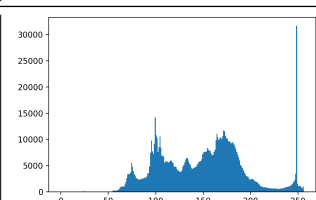
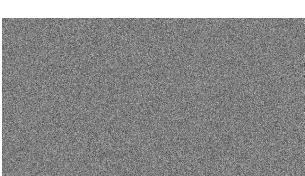
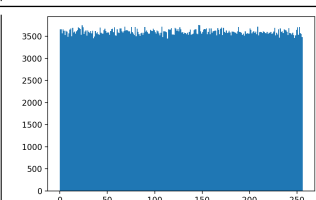

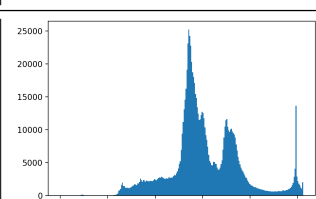
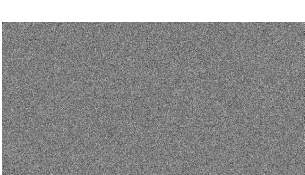
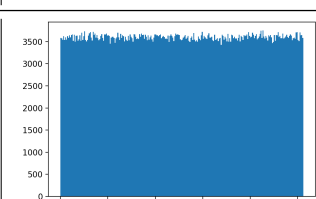
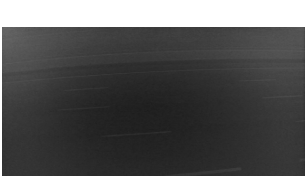
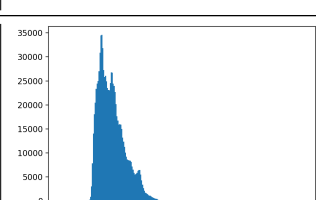
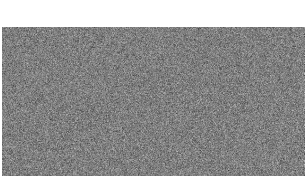
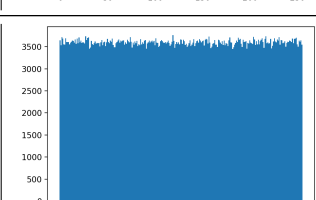
B. KEY SENSITIVITY

A good encryption algorithm that employs proper confusion and diffusion of information will be highly sensitive to the given parameters, namely the secret key. Thus a small difference between keys will yield completely different encrypted outputs and only the exact secret key will be able to decrypt correctly. In Fig. 4, the 256×256 'Lena' has a 99.62% change in pixels of the encrypted output when a 1-bit change is made in the least significant bit of the secret key. In Table 3, the least significant bit of each sub-value of the secret key are changed between encryption and decryption, and show a consistent change in percentage of pixel values. This shows that ChaosNet is highly sensitive to even the smallest change in any of the parameters of the secret key, and show the decrypted image and statistics remain unintelligible with even a marginally incorrect key, proving strong cryptographic properties.

C. LOW ENTROPY ENCRYPTION ANALYSIS

In order to evaluate the algorithm effectiveness at edge cases, the use of an extremely low entropy 256×256 white dot input. Fig. 5.a and Fig. 5.b, and a 1-layer ChaosNet is used to represent the worst case scenario. As can be noted from the figure, the output of ChaosNet is visually and statistically independent of the input. Therefore the algorithm will yield unintelligible encrypted output and flattened histogram and high entropy as depicted in Fig. 5.c and Fig. 5.d, respectively, even with a highly redundant input.

TABLE 2. ITS highway images from the SERSU [2] device using ChaosNet image encryption. Columns 1 and 2 show the plainimage and its 8-bit grayscale histogram respectively. Columns 3 and 4 show the encrypted plainimage using ChaosNet and its associated 8-bit grayscale histogram, respectively.

Plainimage	Plainimage Histogram	Cipherimage	Cipherimage Histogram
			
			
			
			
			
			

D. KEY SPACE ANALYSIS

Every viable encryption scheme should nullify brute force attacks by using a large key space for their algorithms. By having a large key space, the processing time required for a brute force attack increases exponentially for every bit added to the size of the key. For a sufficient level of security the key space should be greater than 2^{128} [30], i.e., the key size should be greater than 128 bits.

A typical block length for block cipher algorithms is 128 bits, thus a 128-bit block will be used as input. Since

the input is a 8-bit grayscale image, the input will technically be $128/8 = 16$ pixels. Using $N = 16$ bytes of inputs, fixed 10-bit lengths $l_\alpha, l_\rho, l_\beta$ of the Lorenz attractor parameters α, ρ, β respectively, the worst case scenario of ChaosNet results in a key length of,

$$\begin{aligned} \text{Key Length} &= (l_\alpha + l_\rho + l_\beta) + 8 \times n \\ &= 10 + 10 + 10 + 8 \times 16 = 158 \text{ bits.} \end{aligned}$$

This worst case scenario shows the key length to be 158 bits implying the key space has 2^{158} possible keys. This shows

TABLE 3. Percent change in pixel values of encrypted image with a 1-bit change in different areas of secret key $[d_\alpha, d_\rho, d_\beta, h_{key}]$ -

Secret Key Sub-Value	Δ Pixel Values (%)
d_α	99.617
d_ρ	99.620
d_β	99.609
h_{key}	99.615

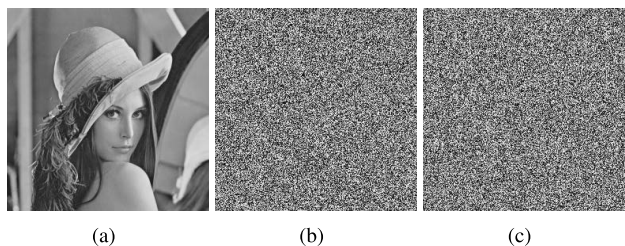


FIGURE 4. Histogram comparison between encryption with K_A and decryption with K_B (K_A with one bit error), where $|K_A - K_B| = 0x1$. (a) Original image. (b) Encrypted image using K_A . (c) Decrypted image using K_B . (d) Histogram of encrypted image using K_A . (e) Histogram of decrypted image using K_B .

sufficient security in terms of brute force attacks since the key space is greater than the aforementioned 2^{128} , and will linearly increase with block size.

E. DIFFERENTIAL ATTACK ANALYSIS

A useful tool for detecting weaknesses from differential attacks is the use of the NPCR and UACI. These metrics

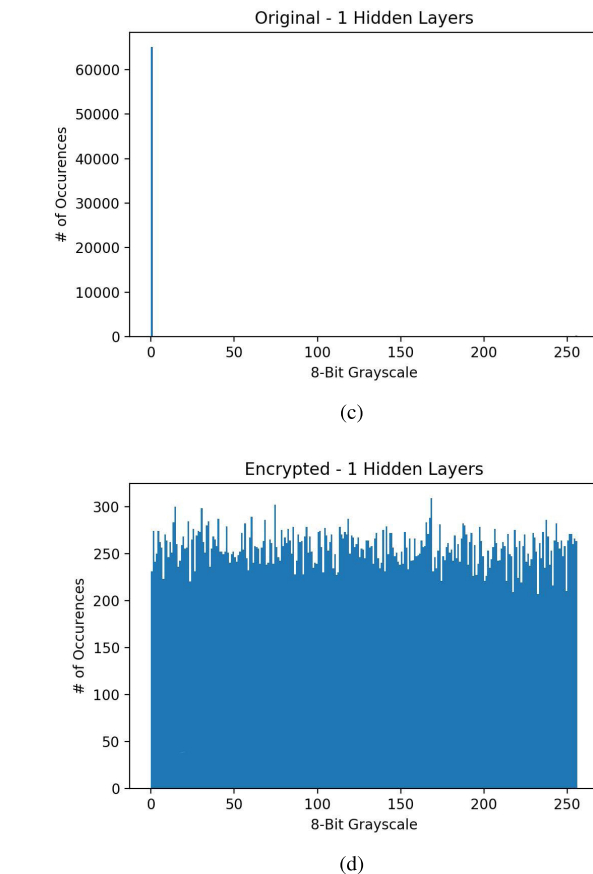
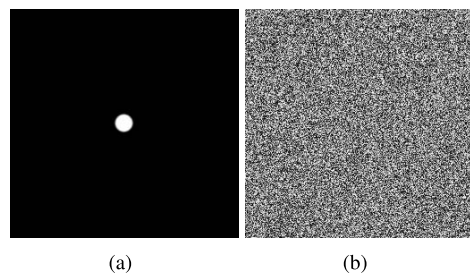


FIGURE 5. Histogram and information of a low entropy image with associated encrypted output using a 1-layer ChaosNet. (a) Original Dot. (b) Dot encrypted with ChaosNet. (c) Original dot histogram. (d) Encrypted dot histogram.

attempt to find a pattern in the encryption algorithm by comparing two cipherimages. A NPCR (9) measure is optimal when it is closest to one, meaning the encryption algorithm has a higher sensitivity to the plainimage. Furthermore the optimal values for UACI (10) is 0.33, which implies that the average change in pixel intensity between encrypted images is 33%,

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

$$D(i,j) = \begin{cases} 1, & C(i,j) \neq C'(i,j) \\ 0, & C(i,j) = C'(i,j) \end{cases} \quad (9)$$

TABLE 4. Comparisons NPCR of other chaos-based encryption algorithms.

Algorithm	NPCR		
	Lena 256×256	Lena 512×512	Cameraman 256×256
[11]	0.9960	-	-
[34]	0.9960	-	-
[35]	-	0.996094	-
[36]	-	0.996036	-
[15]	-	0.9957	-
[37]	-	-	0.9953
ChaosNet	0.99631	0.99630	0.99622

TABLE 5. UACI of several chaos-based encryption algorithms.

Algorithm	UACI		
	Lena 256×256	Lena 512×512	Cameraman 256×256
[11]	0.3357	-	-
[34]	0.3343	-	-
[35]	-	0.306605	-
[36]	-	0.330615	-
[15]	-	0.35082	-
[37]	-	-	0.2688
ChaosNet	0.33337	0.33334	0.33309

$$UACI = \frac{1}{H \times W} \left(\sum_{i,j} \frac{|C(i,j) - C'(i,j)|}{b} \right) \times 100\% \quad (10)$$

where H and W are the image height and width respectively, $C(i, j)$ and $C'(i, j)$ are the values of the different cipherimages at position (i, j) , and b is the range of intensities of the image (for an 8-bit grayscale $b = 255$). In Table 4, the NPCR of the cipherimages is comparable to other algorithms with high NPCR, however it is consistent for each test image, less than 0.002% difference. This adds strength to ChaosNet being resilient to differential analysis attacks. In Table 5, ChaosNet performs the best for each test image (closest to 33.33%) while also remaining consistent through each test image encryption (less than 0.3% difference). Similarly in Table 6, the worst case images i.e. low-resolution grayscale images are encrypted using ChaosNet and show that the NPCR, UACI, and image entropy metrics stay consistently high regardless of the input image.

F. PIXEL CORRELATION

Normal images typically exhibit a certain level of pixel redundancy, since a non-edge segment of an image will be highly correlated to adjacent pixels. A good image encryption algorithm should convincingly decorrelate the pixels' values to improve security, and hence, the correlation should be close to zero. To evaluate the ChaosNet in this regard, 2000 random adjacent pixel positions are used in a horizontal, vertical, and diagonal direction, Fig. 6. The equations used to calculate

TABLE 6. NPCR, UACI, and image entropy of various 256 × 256 8-bit grayscale test images using ChaosNet encryption.

Test Image	NPCR	UACI	Entropy
Baboon	0.99643	0.33500	7.9968
Pepper	0.99620	0.33410	7.9975
Barbara	0.99614	0.33455	7.9975
Castle	0.99648	0.33497	7.9971
Airplane	0.99632	0.33445	7.9974
Monarch	0.99648	0.33313	7.9972
Boat	0.99622	0.33475	7.9975

adjacent pixel $(x$ and $y)$ correlations r_{xy} are,

$$r_{xy} = \frac{N^2 cov(x, y)}{\sum_{i=1}^N (x_i - E_x)^2 \sum_{i=1}^N (y_i - E_y)^2}$$

$$E_x = \frac{\sum_{i=1}^N x_i}{N}, \quad E_y = \frac{\sum_{i=1}^N y_i}{N}$$

$$cov(x, y) = E[(x - E_x)(y - E_y)]. \quad (11)$$

Fig. 6 (a), (c) and (e) show the adjacent pixel correlations of the normal image 'Lena'. As can be noted from the figure, the relationship is linear, which implies that a pixel value at a certain position will have the same or similar pixel value at an adjacent position. Fig. 6. (b), (d) and (f) show the encrypted output of the normal image results in a uniformly scattered pixel correlation, demonstrating that no evidence of leaking information in terms of pixel value patterns and correlation. Table 7 compares the correlation of adjacent pixels to other chaos based algorithms for the specified images, with ChaosNet showing the best results for many of the criteria. It is important to note that in Table 8, encrypting the worst case images i.e. low-resolution test images consistently give very low pixel correlation regardless of the input image, which show the robustness of ChaosNet.

V. OCCLUSION ATTACK ANALYSIS

An occlusion/cropped attack attempts to slightly change the intercepted cipher image by occluding parts of the cipher image in order to render decryption impossible even with the correct key. With the correct key, decryption should still produce recognizable images, though skewed, even with parts of the cipher image gone. In Fig. 7, an occlusion of varied size is applied to a cipherimage, and then is decrypted with the correct secret key. This figure shows that the decrypted image is still recognizable even with 1/9th of the cipherimage gone, showing robustness to cropped attacks. This is also shown in Fig. 8, where different images are still recognizable across different cropped out parts of the cipherimage.

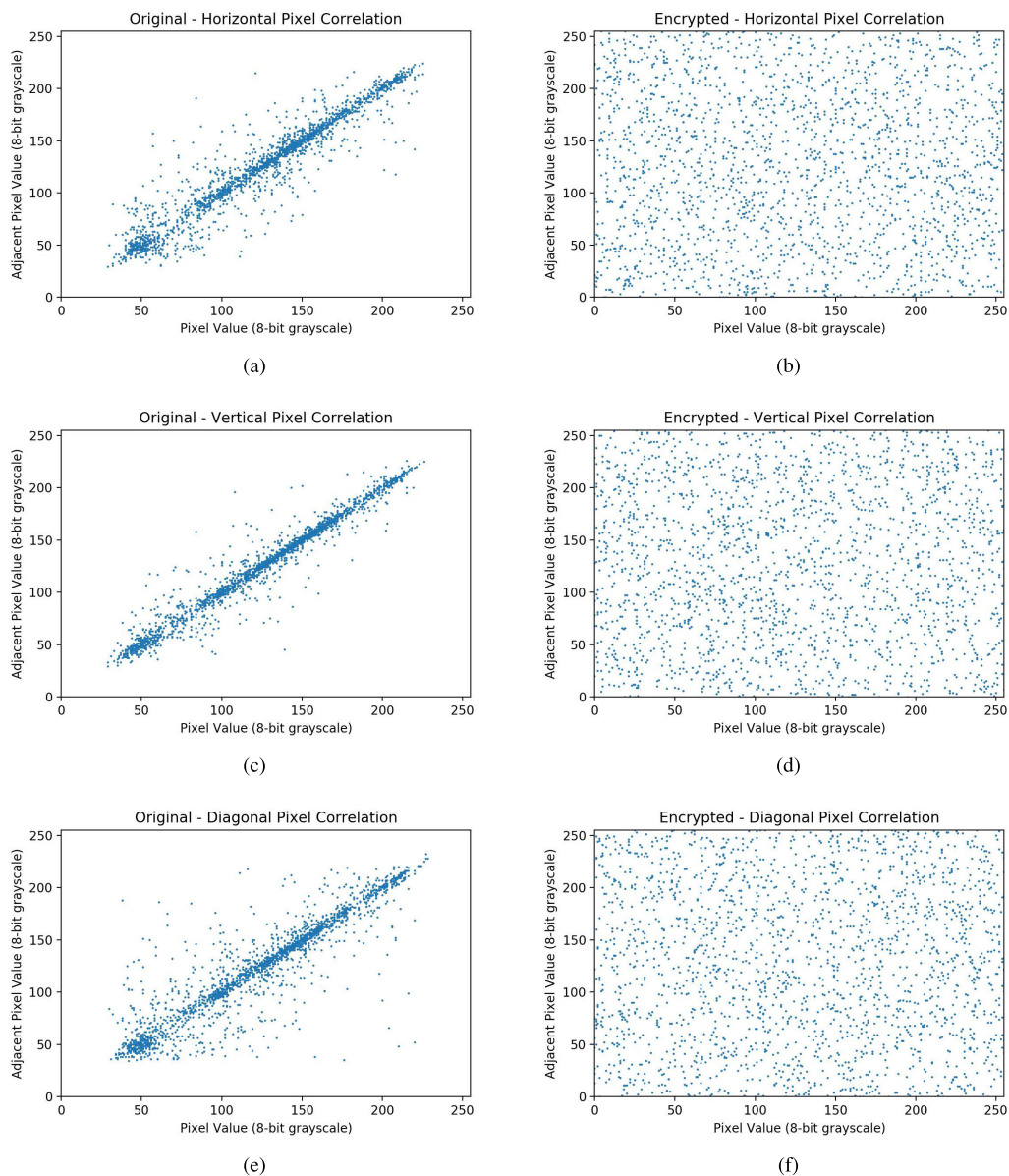


FIGURE 6. Pixel correlation graphs of 2000 random adjacent pixel is specified directions. (a) Horizontal correlation of plain image. (b) Horizontal correlation of cipher image. (c) Vertical correlation of plain image. (d) Vertical correlation of cipher image. (e) Diagonal correlation of plain image. (f) Diagonal correlation of cipher image.

TABLE 7. Comparison of pixel correlation coefficients with other chaotic encryption schemes, using 2000 adjacent pixels in the vertical, horizontal and diagonal directions of the cipherimage.

Algorithm	Correlation Coefficients					
	Lena 256×256			Lena 512×512		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
[15]	0.0014	0.0023	0.0002	-	-	-
[36]	0.0722	0.0099	0.0201	-	-	-
[23]	0.0008	0.0031	0.0014	0.0072	0.0045	0.0033
[10]	-	-	-	0.0055	0.0064	0.0072
[34]	-	-	-	0.0027	0.0152	0.0071
ChaosNet	-0.008131	-0.001382	-0.001016	-0.007393	-0.005329	0.000608

TABLE 8. Pixel correlation of various 256x256 8-bit grayscale test images using ChaosNet encryption.

Test Image	Horizontal	Vertical	Diagonal
Baboon	-0.000577	0.009803	-0.000764
Pepper	-0.004315	-0.007271	-0.000843
Barbara	-0.007070	0.000548	-0.005944
Castle	0.002124	0.000479	0.005367
Airplane	0.009703	-0.008704	-0.004715
Monarch	0.004336	0.000496	0.000835
Boat	0.009503	-0.002979	-0.000418

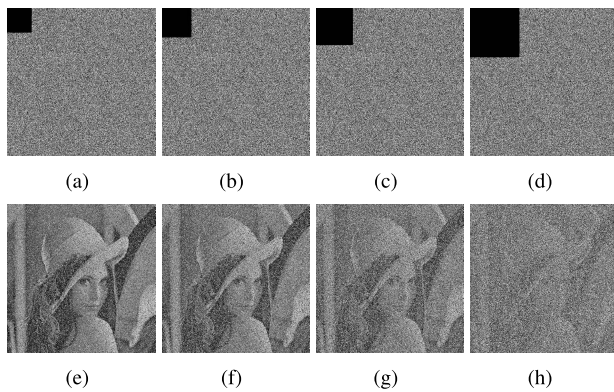


FIGURE 7. Shows the robustness to occlusion/cropped attacks with occlusions of varied sizes. (a) Encrypted image with 1/36th occlusion. (b) Encrypted image with 1/25th occlusion. (c) Encrypted image with 1/16th occlusion. (d) Encrypted image with 1/9th occlusion. (e) Decrypted image of 7(a). (f) Decrypted image of 7(b). (g) Decrypted image of 7(c). (h) Decrypted image of 7(d).

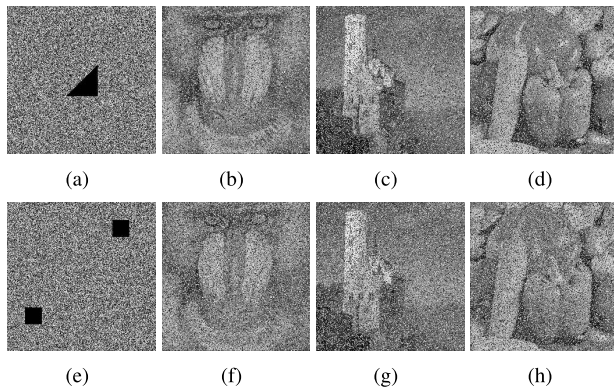


FIGURE 8. Shows the robustness to occlusion/cropped attacks with various images. (a) Occlusion area in encrypted image. (b) Decrypted baboon with 8(a) occlusion. (c) Decrypted castle with 8(a) occlusion. (d) Decrypted peppers with 8(a) occlusion. (e) Occlusion area in encrypted image. (f) Decrypted baboon with 8(e) occlusion. (g) Decrypted castle with 8(e) occlusion. (h) Decrypted peppers with 8(e) occlusion.

VI. CONCLUSION

This paper has presented an image encryption scheme for ITS applications, ChaosNet, based on the Lorenz chaotic system and key controlled neural networks with finite field weight matrices. The obtained results show that the proposed ChaosNet system has superior cryptographic properties, where it

outperformed the other considered chaotic image encryption algorithms in most tests, which included information entropy, key sensitivity, low entropy encryption analysis, differential attack analysis, pixel correlation, and occlusion attack analysis. Therefore, the proposed system can be considered attractive for applications with sensitive data such as ITSs. Future work includes algorithm optimization and hardware implementation for increased performance.

REFERENCES

- [1] A. Al-Dweik, M. Mayhew, R. Muresan, S. M. Ali, and A. Shami, "Using technology to make roads safer: Adaptive speed limits for an intelligent transportation system," *IEEE Veh. Technol. Mag.*, vol. 12, no. 1, pp. 39–47, Mar. 2017.
- [2] A. Al-Dweik, R. Muresan, M. Mayhew, and M. Lieberman, "IoT-based multifunctional scalable real-time enhanced road side unit for intelligent transportation systems," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr./May 2017, pp. 1–6.
- [3] X. Wu, B. Zhu, Y. Hu, and Y. Ran, "A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps," *IEEE Access*, vol. 5, pp. 6429–6436, 2017.
- [4] G. Thoms, R. Muresan, and A. Al-Dweik, "Design of chaotic block cipher operation mode for intelligent transportation systems," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–4.
- [5] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, "A novel design of Chaos based S-boxes using genetic algorithm techniques," in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 678–684.
- [6] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.
- [7] M. Khan, T. Shah, and S. Batool, "Construction of S-box based on chaotic Boolean functions and its application in image encryption," *Neural Comput. Appl.*, vol. 27, no. 3, pp. 677–685, 2016.
- [8] P. Praveenkumar, R. Amirtharajan, K. Thenmozhi, and J. B. B. Rayappan, "Triple chaotic image scrambling on RGB—A random image encryption approach," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 3335–3345, 2015.
- [9] X.-J. Tong, M. Zhang, Z. Wang, and J. Ma, "A joint color image encryption and compression scheme based on hyper-chaotic system," *Nonlinear Dyn.*, vol. 84, no. 4, pp. 2333–2356, 2016.
- [10] L. Wang, H. Song, and P. Liu, "A novel hybrid color image encryption algorithm using two complex chaotic systems," *Opt. Lasers Eng.*, vol. 77, pp. 118–125, Feb. 2016.
- [11] X. Wang and H.-L. Zhang, "A novel image encryption algorithm based on genetic recombination and hyper-chaotic systems," *Nonlinear Dyn.*, vol. 83, nos. 1–2, pp. 333–346, 2016.
- [12] L. Xu, Z. Li, J. Li, and W. Hua, "A novel bit-level image encryption algorithm based on chaotic maps," *Opt. Lasers Eng.*, vol. 78, pp. 17–25, Mar. 2016.
- [13] A. Y. Niyat, M. H. Moattar, and M. N. Torshiz, "Color image encryption based on hybrid hyper-chaotic system and cellular automata," *Opt. Lasers Eng.*, vol. 90, pp. 225–237, Mar. 2017.
- [14] J. Zhang, "An image encryption scheme based on cat map and hyper-chaotic lorenz system," in *Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol.*, Feb. 2015, pp. 78–82.
- [15] Y. Abanda and A. Tiedeu, "Image encryption by chaos mixing," *IET Image Process.*, vol. 10, no. 10, pp. 742–750, Oct. 2016.
- [16] H. Diab, "An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations," *IEEE Access*, vol. 6, pp. 42227–42244, 2018.
- [17] X.-Q. Fu, B.-C. Liu, Y.-Y. Xie, W. Li, and Y. Liu, "Image encryption-then-transmission using DNA encryption algorithm and the double chaos," *IEEE Photon. J.*, vol. 10, no. 3, Jun. 2018, Art. no. 3900515.
- [18] S. Ma, Y. Zhang, Z. Yang, J. Hu, and X. Lei, "A new plaintext-related image encryption scheme based on chaotic sequence," *IEEE Access*, vol. 7, pp. 30344–30360, 2019.
- [19] C. Zhu and K. Sun, "Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps," *IEEE Access*, vol. 6, pp. 18759–18770, 2018.

- [20] N. Zhou, A. Zhang, F. Zheng, and L. Gong, "Novel image compression-encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing," *Opt. Laser Technol.*, vol. 62, pp. 152–160, Oct. 2014.
- [21] J. Fridrich, "Image encryption based on chaotic maps," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Simulation*, vol. 2, Oct. 1997, pp. 1105–1110.
- [22] E. Yavuz, R. Yazici, M. C. Kasapbaşı, and E. Yamaç, "A chaos-based image encryption algorithm with simple logical functions," *Comput. Electr. Eng.*, vol. 54, pp. 471–483, Aug. 2016.
- [23] A. Boutros, S. Hesham, B. Georgey, and M. A. A. El Ghany, "Hardware acceleration of novel chaos-based image encryption for IoT applications," in *Proc. 29th Int. Conf. Microelectron. (ICM)*, Dec. 2017, pp. 1–4.
- [24] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An optical image compression and encryption scheme based on compressive sensing and RSA algorithm," *Opt. Laser Eng.*, vol. 121, pp. 169–180, Oct. 2019.
- [25] S.-S. Yu, N.-R. Zhou, L.-H. Gong, and Z. Nie, "Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system," *Opt. Laser Eng.*, vol. 124, Jan. 2019, Art. no. 105816.
- [26] N. Thein, H. A. Nugroho, T. B. Adji, and I. W. Mustika, "Comparative performance study on ordinary and chaos image encryption schemes," in *Proc. Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov/Dec. 2017, pp. 122–126.
- [27] G. P. Williams, *Chaos Theory Tamed*. Washington, DC, USA: Joseph Henry Press, 1997.
- [28] J. C. Sprott, *Chaos and Time-Series Analysis*. London, U.K.: Oxford Univ. Press, 2006.
- [29] W. Stallings, *Cryptography and Network Security*. London, U.K.: Pearson, 2017.
- [30] *Federal Information Processing Standards Publication 197, Announcing the Advanced Encryption Standard (AES)*, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2001.
- [31] Z.-J. Huang, S. Cheng, L.-H. Gong, and N.-R. Zhou, "Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform," *Opt. Laser Eng.*, vol. 124, Jan. 2020, Art. no. 105821.
- [32] N. Zhou, H. Jiang, L. Gong, and X. Xie, "Double-image compression and encryption algorithm based on co-sparse representation and random pixel exchanging," *Opt. Lasers Eng.*, vol. 110, pp. 72–79, Nov. 2018.
- [33] J. V. Stone, *Information Theory: A Tutorial Introduction*. Sebtel Press, 2015.
- [34] L. Kong and L. Li, "A new image encryption algorithm based on chaos," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 4932–4937.
- [35] N. Kumar et al., "Chaos and hill cipher based image encryption for mammography images," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Mar. 2015, pp. 1–5.
- [36] S. S. Alam, S. Bhattacharyya, and S. Chandra, "A novel image encryption algorithm using hyper-chaos key sequences, multi step group based binary gray conversion and circular bit shifting logic," in *Proc. Int. Conf. Sci. Eng. Manage. Res. (ICSEMR)*, Nov. 2014, pp. 1–9.
- [37] M. G. Avasare and V. V. Kelkar, "Image encryption using chaos theory," in *Proc. Int. Conf. Commun., Inf. Comput. Technol. (ICCICT)*, Jan. 2015, pp. 1–6.
- [38] H. Zhu, C. Zhao, and X. Zhang, "A novel image encryption-compression scheme using hyper-chaos and Chinese remainder theorem," *Signal Process., Image Commun.*, vol. 28, no. 6, pp. 670–680, 2013.
- [39] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 514–522, 2011.
- [40] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft. Comput.*, vol. 12, no. 5, pp. 1457–1466, 2012.
- [41] A. H. Abdullah, R. Enayatifa, and M. Lee, "A hybrid genetic algorithm and chaotic function model for image encryption," *AEU-Int. J. Electron. Commun.*, vol. 66, no. 10, pp. 806–816, 2012.
- [42] S. Bhowmik and S. Acharyya, "Image cryptography: The genetic algorithm approach," in *Proc. IEEE Int. Conf. Comput. Sci. Automat. Eng.*, vol. 2, Jun. 2011, pp. 223–227.
- [43] R. Afarin and S. Mozaffari, "Image encryption using genetic algorithm," in *Proc. 8th Iranian Conf. Mach. Vis. Image Process. (MVIP)*, Sep. 2013, pp. 441–445.



GRAHAM R. W. THOMS (S'19) received the B.Eng. and M.A.Sc. degrees in computer engineering from the University of Guelph, Canada, in 2017 and 2019, respectively. He is currently engaging in industry. His current research interests include image processing, machine learning, cryptography, real-time embedded systems, chaotic systems, and intelligent transportation systems.



RADU MURESAN received the M.A.Sc. and Ph.D. degrees from the University of Waterloo, Canada, in 2001 and 2003, respectively, all in electrical and computer engineering. He is currently an Associate Professor with the Engineering Department, University of Guelph, Canada. His current research interests include VLSI design, system-on-chip design, security, and intelligent embedded systems design. In the area of security, he studies the integration of highly secure cryptographic components into intelligent embedded systems. Specifically, the design and integration of on-chip countermeasure circuits against side-channel attacks, physically unclonable function modules, and chaotic ciphers. He is an Ontario P.Eng. and a member of the IEEE Circuits and System Society, the IEEE Solid-State Circuits Society, and ACM organizations.



ARAFAT AL-DWEIK (S'97–M'01–SM'04) received the B.Sc. degree in telecommunication engineering from Yarmouk University, Jordan, in 1994, and the M.S. (*summa cum laude*) and Ph.D. (*magna cum laude*) degrees in electrical engineering from Cleveland State University, Cleveland, OH, USA, in 1998 and 2001, respectively. He was with Efficient Channel Coding Inc., Cleveland, from 1999 to 2001, where he was a Research and Development Engineer involved in advanced modulation, coding, and synchronization techniques. From 2001 to 2003, he was the Head of the Department of Information Technology, Arab American University, Palestine. From 2003 to 2012, he was with the Communications Engineering Department, Khalifa University, United Arab Emirates. He joined the University of Guelph, Guelph, ON, Canada, as an Associate Professor, from 2013 to 2014. He has been a Visiting Research Fellow with the School of Electrical, Electronic and Computer Engineering, Newcastle University, Newcastle upon Tyne, U.K., since 2006. He is currently a Research Professor and a Member of the School of Graduate Studies, Western University, London, ON, Canada. He has authored over 110 published articles and five issued U.S. patents. He has received several research awards and was a recipient of the Fulbright Scholarship. He has extensive editorial experience, where he serves as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and *IET Communications*.

• • •