Wireless Sensor Networks Encryption Scheme Using Chaotic Map and Genetic Operations

R.Sarathkumar Kithiyon[1] C.Karuppasami[2] S.Immanuel Prabaharan[3] A.Samsu Nighar[4]
[1,2]B.E. Student [3,4]Assistant Professor
[1,2,3,4]Department of Electronics and Communication Engineering
[1,2,3,4]Chandy College of Engineering, Tuticorin, Tamil Nadu, India

Abstract— Over the past decade, the application domain of wireless sensor networks has expanded steadily, ranging from environmental management to industry control, and from structural health monitoring to strategic surveillance. With the proliferation of sensor networks at home, work place, and beyond, securing data in the network has become a challenge. A number of security mechanisms have been proposed for sensor networks to provide data confidentiality: 1) advanced encryption system; 2) KATAN; 3) LED; and 4) TWINE. However, these schemes have drawbacks, including security vulnerabilities, need for hardware based implementation, and higher computational complexity. To address these limitations, we propose a lightweight block cipher based on chaotic map and genetic operations. The proposed cryptographic scheme employs elliptic curve points to verify the communicating nodes and as one of the chaotic map parameters to generate the pseudorandom bit sequence. This sequence is used in XOR, mutation, and crossover operations in order to encrypt the data blocks. The experimental results based on Mica2 sensor mote show that the proposed encryption scheme is nine times faster than the LED protocol and two times faster than the TWINE protocol. We have also performed a number of statistical tests and cryptanalytic attacks to evaluate the security strength of the algorithm and found the cipher provably secure.

Key words: Wireless sensor network, pseudorandom bit sequence generator, data encryption, elliptic curve, chaotic map, mutation, crossover

I. INTRODUCTION

WIRELESS Sensor Networks (WSNs) have experienced a rapid growth during the last few years. The continuous advancement in wireless technologies, intelligent sensors and micro-electronic- mechanical systems (MEMS) has increased the scope of application domains of sensor networks. WSNs aimed at various industrial, medical, and military applications necessitate research in the design of secure and energy efficient protocols. Specially, the use of sensors in critical systems such as nuclear power plants, aircrafts, and hospitals requires effective mechanisms to ensure the authenticity, confidentiality and integrity of the sensed and transmitted data [1]. However, security is a challenging issue in WSNs, since sensors are usually deployed in hostile environments. Moreover, limited memory and processing power and short communication range of sensor nodes (SNs) introduce several challenges when implementing traditional cryptographic schemes in wireless environments. WSNs thus require efficient encryption schemes in terms of storage space, power consumption, and operating speed. The key issue in designing crypto-systems for WSNs is to maintain the trade-off among security, performance and cost. Several encryption algorithms for resource constrained SNs have been designed in the past few years. These algorithms may be classified into three main categories: compact hardware oriented cryptographic schemes, conventional block ciphers, and lightweight block ciphers. Highly optimized and compact block ciphers (e.g., KATAN and KTANTAN) are not readily suitable for WSNs, since the energy consumption and memory usage are high [2]. On the other hand, most of the classical block ciphers adopted for WSNs are vulnerable to a number of security attacks [5], [7]. Therefore, the current research focuses on designing secure and lightweight block ciphers. In spite of the best efforts of researchers, many of these light- weight ciphers have relatively poor performance compared to conventional cryptographic schemes. For example, the number of CPU cycles to encrypt one byte data in conventional crypto- systems (e.g., Tiny Encryption Algorithm (TEA) and extended TEA) is less than 2000, whereas the lightweight block ciphers (e.g., LED and TWINE) require about 5500 cycles [2]. To address these shortcomings, we propose a chaotic map and genetic operations based block cipher for tiny sensor devices enabling low cost and secure data communication between the source and the destination nodes.

The proposed scheme includes a number of benefits: i) it uses the discrete chaotic map, which supports a wider data range with low computational cost. Most of the encryption schemes use fixed chaotic map parameters to generate the random bit sequences, but our algorithm uses random values of ‘x’ and ‘y’ for every session generated by elliptic curve operations; ii) the proposed crypto-system makes different pseudorandom bit sequences for every session and thus preserves independent behavioural characteristics of the algorithm; iii) the scheme is more efficient compared to Skip Jack, Advanced Encryption System (AES), LED, TWINE, and Block Cipher based on Chaos (BCC) in terms of CPU consumption and encryption time; iv) the proposed encryption algorithm is suitable for both text and image encryption. From the application point of view, it is desirable for the crypto-system to protect confidential information not only in text form but also in image form. Image data differ from text due to intrinsic features, such as strong correlation between adjacent pixels and high redundancy. Hence, the encryption scheme should be robust, fast, and computationally secure. Experimental results show that the proposed block cipher ensures all these properties. The main contributions of this paper are threefold.

A new key establishment procedure, which minimizes the implementation gap between different security mechanisms. This is achieved by employing the same elliptic curve points for both the node-verification and pseudo random bit sequence generation processes.

A novel cryptographic scheme that integrates the benefits of elliptic curve, discrete chaotic map, and genetic cryptography for WSN applications. This integration ensures an adequate level of security with limited resources.

A robust block cipher that can be used for both text and image data encryption. It enables the use of the same encryption mechanism in multi-mode sensors, for example, sensing different environmental phenomena as well as images

The rest of the paper is organized as follows: In Section II, we discuss the feasibility of existing security mechanisms in WSN environments. Section III provides details of the new key establishment procedure, pseudorandom bit sequence generation process and the proposed encryption scheme. A concrete security treatment and analysis is presented in Section IV. Section V and Section VI examine the security and performance analysis of our proposed protocol respectively. Finally, Section VII concludes the paper.

II. RELATED WORKS

We critically examine a number of existing security mechanisms and their suitability for WSNs. Here, we provide an overview of a number of security protocols used in WSN applications.

RC5 is a flexible block cipher that has a variable block size (32, 64, or 128 bits), number of rounds (0-255), and key size (0-2040 bits). Although RC5 is considered more suitable for WSNs, the key scheduling process increases both memory and computational costs [3]. Moreover, the RC5 cipher is designed to take advantage of variable-bit rotation instruction (e.g., ROL), which is not supported by many embedded systems like Intel architecture [4].

Another widely used block cipher in WSN is Skipjack developed by the US National Security Agency (NSA). It uses an 80-bits key to encrypt or decrypt 64-bit data blocks. The short key length makes Skip Jack vulnerable to the exhaustive key search attack [5]. An extended version, Skip Jack-X is proposed by Sen Sec designers to make the cipher more secure against security attacks. However, it is seen that the strategy is not a proper replacement of Skip Jack in WSNs.

Tiny Encryption Algorithm (TEA) is notable for its simple structure and small memory requirement. It has a few weaknesses, such as being vulnerable to a related-key attack and chosen plaintext attack [6]. To eliminate these weaknesses, a Corrected Block TEA (XXTEA) is designed with128-bits key. However, the last reported attack against full-round XXTEA presents a chosen plaintext attack using 2^{59} queries and negligible work [7].

The Advanced Encryption System (AES) algorithm is a widely used block cipher based on a substitution-permutation process and has a fixed block size of 128 bits. It operates on a 4 × 4 array of bytes and has a key size of 128, 192, or 256 bits. However, AES running on 10, 12, and 14 rounds for 128, 192, and 256-bits key respectively is still found vulnerable by the researchers [8]. In addition to security issues, AES is mainly not suitable for WSNs due to the demand for more hardware resources [9].

KATAN and KTANTAN are two block ciphers proposed by Cannier et al. [10]. Both ciphers use blocks of sizes 32, 48 or 64 bits under 80 bits key and iterate for 254 rounds. The main difference between KATAN and KTANTAN is the key scheduling scheme. The 80 bits key in KATAN is loaded into a register and is repeatedly clocked, whereas in KTANTAN the key is fixed. These two encryption schemes are vulnerable to a number of security attacks. A conditional differential cryptanalysis with a practical complexity in single key settings and related key settings is presented against KATAN [11], [12]. Similarly, a meet-in-the-middle attack is proposed against KTANTAN that recovers the 80-bits secret key of the full rounds KTANTAN (32/48/64 bits) at time complexity of $2^{72.9}$, $2^{73.8}$, and $2^{74.4}$[13]. Furthermore, these two algorithms are expensive in terms of energy and memory consumption.

LED is a lightweight block cipher that encrypts 64 bits blocks using either 64 bits or 128 bits key with 32 or 48 rounds respectively [14]. Instead of key scheduling, the key is XORed at every four rounds in LED. This feature is compensated by a larger number of rounds compared to the AES. A meet-in-the-middle attack against 8 rounds of LED-64 and 16 rounds of LED-128 and the results of differential cryptanalysis of full LED in the related key settings are presented in [15] and [16]. Besides these pitfalls, the LED cipher also consumes more CPU cycles compared to conventional cryptographic schemes.

TWINE is a 64 bits block cipher that uses an 80 bits or 128 bits key [17]. It employs a generalized feistel structure with 16 branches and iterates for 36 rounds. The internal F-function is repeated 8 times in each round and is composed of a sub-key addition and a single S-box. The best known attacks against TWINE are two biclique attacks on TWINE-80 and TWINE-128 with the time complexities equal to $2^{79.1}$and $2^{126.8}$ respectively, with a data requirement for the two attacks equal to 2^{60} [18].

The Simple Lightweight Encryption Scheme (SLES) is a block cipher that uses elliptic curve operations over prime field to generate pseudorandom bit sequences [19]. Instead of using a fixed base point for the entire lifetime of a WSN, SLES pregenerates a large key pool to share a new key at the beginning of the communication process. This key is used as a new base point to generate the random bit sequence. The limitation of SLES is that the computational time increases when the range of elliptic curve parameters is extended.

The chaos-based cryptographic scheme has been widely investigated over the past few years and a number of chaotic block ciphers are proposed for conventional networks. However, most of these chaos-based ciphers are not suitable for WSNs, since sensor nodes are resource constrained. The chaotic maps usually generate the sequences of random floating-point numbers which are not supported by tiny sensor nodes (e.g., Mica2). To overcome this problem, a chaos block cipher with integer parameters was proposed [20]. The algorithm divides the plaintext into 8-bit blocks and then performs bit permutation. The permuted bits are encrypted in four rounds Feistel cipher using four bytes sub-keys. These sub-keys are generated by performing XOR operations with 32-bits integer chaos. Finally, the 8-bits are permuted again to generate the corresponding ciphertext. However, this chaos block cipher cannot resist differential attack because the number of rounds is too small and the calculation precision of round

function is too short [21]. Moreover, the energy consumption and memory usage are also not analysed in this protocol.

Xiao-Jun et al. proposed a fast, secure, and low resource consumption algorithm based on the integer discretization of a chaotic map, the Feistel network structure and an S-box [9]. The encryption algorithm uses a block length of 32 bits, a key length of 128 bits, an initial vector of 32 bits, and 14 rounds iteration to generate 32 bits ciphertext. Experimental outcomes and analysis show that the cipher has a large key space, very good diffusion and good statistical balance. The main drawback of this protocol is that 32-bit block size is not semantically secure to resist a chosen-plaintext attack.

Another block cipher based on chaotic S-boxes and a substitution-permutation network was proposed in [22]. The encryption procedure avoids floating-point operations and multiplications in order to minimize the energy consumption. Similar to RC5, the block cipher based on chaos (BCC) supports variable word size (w), number of rounds (r), and length of the encryption key (b). Although the authors claim that BCC has good diffusion property and low energy consumption, they failed to present any security and performance analysis.

III. THE PROPOSED BLOCK CIPHER

Our proposed block cipher is divided into three phases: a) key establishment phase, b) pseudorandom bit sequence generation phase, and c) encryption phase. Each of the phases is described in detail below.

A. Key Establishment Phase

In this phase, a secret key is randomly selected from the key pool and exchanged between sending and receiving nodes. The key establishment phase uses an elliptic curve over prime field to generate a large key pool for node-verification purpose. An elliptic curve over prime field is an algebraic expression and is defined by the following equation:

$$y^2 (\text{mod } p) = x^3 + Ax + B (\text{mod } p) \quad (1)$$

where, A and B are the coefficients and the variables x and y take the values only from the finite field within the range of prime field p . Given the values of these parameters, a large number of points on the curve can be generated using basic elliptic curve operations, known as point addition and point doubling [23].

We assume that the elliptic curve parameters (i.e., prime field p, base point G(x,y), coefficients A and B), and chaotic map parameters (i.e., m, N, μ and β) are predistributed securely among all sensor nodes in the WSN. Now, each SN generates a list of elliptic curve points referred to as key pool by using elliptic curve operations. When a node is required to send data packets, it randomly selects a secret key (xi,yi) from the key pool and converts it into hash code using a pre-defined hash function. Then, the hash code is shared with the destination node. The destination node retrieves the shared key by matching the received code with the hash code generated for each point of its own key pool. Upon successful retrieval of the secret key, destination node verifies the legitimacy of the source node and sends an acknowledgement. This secret key (xi,yi) is used in N-logistic tent map with other parameters to generate the random bit sequence.

B. Generation of Pseudorandom Bit Sequence

This phase involves the generation of pseudorandom bit sequences using chaotic functions. The security level of a discrete chaotic map depends on the properties of the random number generation scheme such as unpredictability and unlimited period. However, most of the chaotic maps involve high- precision floating point calculation to produce the sequences of random floating-point numbers which are not suitable for resource limited SNs. However, the advantage of using N-logistic tent map is that it can deal with integer parameters and thus simplifies the computation process in SNs. We have investigated the randomness of the derived binary sequence using the test code developed by the National Institute of Standards and Technology (NIST) and found that the sequence is random [24]. The following equations present the chaotic functions used to generate the pseudorandom bit sequences in our proposed encryption scheme.

$$\begin{cases} x_{n+1} = \mu x_n (N - x_n/m)/N - y_n/2 \\ y_{n+1} = \beta (N - |N - y_n|) \end{cases} \quad (2)$$

where $x \in (0, m \times N)$, $\mu \in [0, 4]$, $y \in (0, 2 \times N)$, $\beta \in [1, 2]$, $N = 2^K$, and $m = 2^k$ with integers K and k [25]. The seed key is the set $\{x_i, y_i, m, N, \mu, \beta\}$, where, the values of m, N, μ, and β are predistributed in the sensor nodes, and the initial values of x_i and y_i are exchanged through the key establishment phase as explained earlier.

C. The Encryption Process

The overall encryption process is shown as a block diagram presented in Fig. 1 where the symbols 'M' and 'XO' denote mutation and crossover operation respectively. Confusion and diffusion are two general principles that guide the design of a block cipher. Confusion is achieved by obscuring the relationship between the cipher text and the symmetric key as best as possible. On the other hand, diffusion is achieved by dissipating the redundancy of the plaintext through spreading it over the cipher text. The proposed cryptographic scheme implements three different operations: XOR, mutation, and crossover. The additive cipher XOR is secure when the key- stream is as long as the plaintext. On the other hand, mutation is a process of flipping one or multiple bits in a given bit string. Crossover is a process of taking two parent bit strings and producing corresponding child bit strings by interchanging selected parts of bit strings between the parents. These two genetic operations are used in genetic algorithm to generate a new population from the existing one [26]. In our proposed encryption mechanism, the mutation and crossover genetic operations are used as tools for introducing diffusion and confusion properties in the ciphertext.

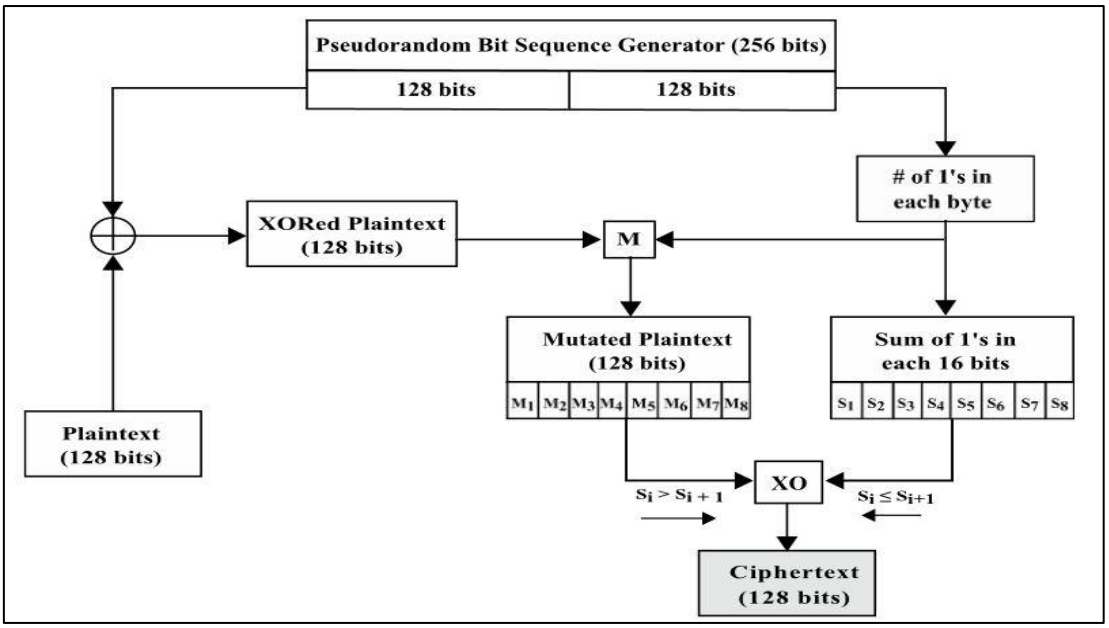

Fig. 1: The general schema of the proposed encryption process.

The mutation process is applied to create random diversity (diffusion) in the ciphertext, whereas the crossover operation is used to change the order of the mutated text or image data (confusion). The main benefit of using genetic operations is that they introduce relatively fair diversity in the ciphertext. Below, we describe the encryption procedure with typical examples

We first divide the pseudorandom bit sequence generated by chaotic map into 256-bit blocks, and each block is divided into two sub-blocks of 128 bits. Then, we calculate the number of 1's in each byte as well as the sum of 1's for each two consecutive bytes in the other half of the pseudorandom bit sequence. After that, we convert the plaintext into their corresponding binary codes and group them into blocks of 128-bits. This block is XORed with the first sub-block of the pseudorandom bit sequence. Then, the mutation is performed on each byte of the XORed binary codes using the total number of 1's in each byte as the starting index of the mutation process. For example, if the number of 1's in the first byte of the second sub-block of a pseudorandom bit sequence is 7, we mutate the 7^{th} and 8^{th} number bits in the first byte of the XORed plaintext. Thereafter, the crossover operation is executed on mutated plaintext as shown in the Fig. 1 to generate the ciphertext. At this step, we take four consecutive bytes from mutated plaintext and then perform crossover operations according to the number of 1's in the second sub-block of pseudorandom bit sequence. For example, in the case of B1–B2–B3–B4 being the four successive bytes of mutated binary codes. We compare the sum of 1's (sum1) in the first two bytes in pseudorandom bit sequence with that of the next two consecutive bytes (sum2). If sum1 is greater than sum2, then the crossover is performed from left to right (i.e., 1 to sum1), otherwise it is done from right to left (i.e., 32 to sum2). This crossover operation is done repeatedly (e.g.,B1–B2–B3–B4,B2–B3–B4–B5,...,B14–B15–B16–1) so that each byte in mutated plaintext performs the operation at least twice. The decryption process is simply the inverse of the encryption procedure.

IV. CONCRETE SECURITY EVALUATION

In this section, we present the formal description of the proposed cryptographic scheme, mathematicalnotation of in distinguishability under a chosen-plaintext attack (IND-CPA), and IND-CPA security analysis.

A. Formal Description

A symmetric encryption algorithm is defined by a family of functions such as F : Keys (F) × Dom(F) → Range(F). For K ∈ Keys(F), FK : Dom(F)→Range(F) can be defined as ∀ x ∈ Dom(F) : FK(x)=F(K, x). Thus, our proposed encryption scheme with Keys(F)=$\{0,1\}^{256}$ and Dom(F)=Range(F)=$\{0,1\}^{128}$ can be expressed as, F:$\{0,1\}^{256}\times\{0,1\}^{128}\rightarrow\{0,1\}^{128}$ where, the mode of operation over F with a random starting point is a stateless block cipher. The encryption and decryption algorithms are presented in Algorithm 1. The starting point S[0] is used to define a set of values on which FK is applied to generate a "pseudorandom bit sequence" of desired length. Then, the sequence is subdivided into two parts (P[i]L and P[i]R) to perform XOR(⊕), mutation(μ) and crossover(⊗) operation as shown in the Algorithm 1

Algorithm 1 Encryption and Decryption Algorithm

1: **Procedure** ENC_ALG $\xi_K(M)$	1: **Procedure** DEC_ALG $D_K(M)$
2: $M[1]\ldots M[n] \leftarrow M$	2: $C[1]\ldots C[n] \leftarrow C$
3: $S[0] \leftarrow \{0,1\}^l$	3: $S[0] \leftarrow \{0,1\}^l$
4: **for** $i = 1\ldots n$ do	4: **for** $i = 1\ldots n$ do
5: $P[i] \leftarrow F_K(S[0], i)$	5: $P[i] \leftarrow F_K(S[0], i)$
6: $CT1[i] \leftarrow P[i]^L \oplus M[i]$	6: $M1[i] \leftarrow C[i] \otimes P[i]^R$
7: $CT2[i] \leftarrow CT1[i]\ \bar{\mu}\ P[i]^R$	7: $M2[i] \leftarrow M1[i]\ \bar{\mu}\ P[i]^R$
8: $C[i] \leftarrow CT2[i] \otimes P[i]^R$	8: $M[i] \leftarrow P[i]^L \oplus M2[i]$
9: **end for**	9: **end for**
10: **return** C	10: **return** M
11. **end procedure**	11. **end procedure**

Table 1:

B. Indistinguishability under Chosen-Plaintext Attack

Suppose an adversary is given a sequence of ciphertexts (C1...Cn) where Ci is either an encryption of M0, i or M1, I

for all 1≤ i ≤n. Now, the goal of the adversary is to generate the sequence of ciphertexts and guess whether M0, i...M0, n were encrypted or M1, i ...M1, n were encrypted. The encryption scheme is considered secure if the adversary finds it hard to distinguish which of the two message sequences correspond to the cipher, Ci.

V. SECURITY ANALYSIS AND TEST RESULTS

We have tested our proposed encryption scheme against various security attacks. Here, we describe some of the important security analysis results including key-space analysis, statistical analysis, differential attack analysis, information entropy analysis, known-plaintext and ciphertext-only attack analysis. The experiments are performed on two gray-level images with a size of 128×128 using MATLAB simulator.

A. Key-Space Analysis

Key space size is determined by the total number of different keys used in the encryption scheme. The key-space of a good encryption algorithm should be large enough to make brute- force attacks infeasible. In our proposed encryption scheme, the set of secret parameters is {xi, yi, μ,β,m, N}, where, xi, yi, μ,β,m, and N are integer values, having the following range:{xi, yi, μ,β, N}∈[1, 2128]and m∈[1, 264]. Therefore, he complete key-space of the proposed encryption scheme is5.087×10135≈2448. Hence, we conclude that brute force attack is not feasible for such a large key space.

B. Statistical Analysis

We have performed several statistical analyses to test the robustness of our proposed encryption procedure. From the experiments, we have found that the encrypted information in the cipher image is nearly uniformly distributed. Here, we present the results of the histogram analysis and correlation analysis of two adjacent pixels in cipher images.

1) Histogram Analysis

An image histogram plots the pixels at each gray scale level in order to illustrate the distribution of pixels in that image. It is expected that the distribution of pixels in the cipher image should hide the redundancy of the original image and should not leak any information about the plain image or the relationship between the plain image and the cipher image. We have calculated and analysed the histograms of two encrypted images as well as original images consisting of different contents. The histograms of the plain images shown in Fig. 2(a) and 2(e) and their corresponding cipher images Fig. 2(c) and 2(g) produced by the proposed scheme are depicted in Fig. 2(b), 2(f); and 2(d), 2(h) respectively. From Fig. 2(d) and 2(h), it can be seen that the histograms of the cipher images are significantly different from that of the plain images and have a reasonably uniform distribution. It demonstrates that the proposed crypto-system can resist statistical attack well.

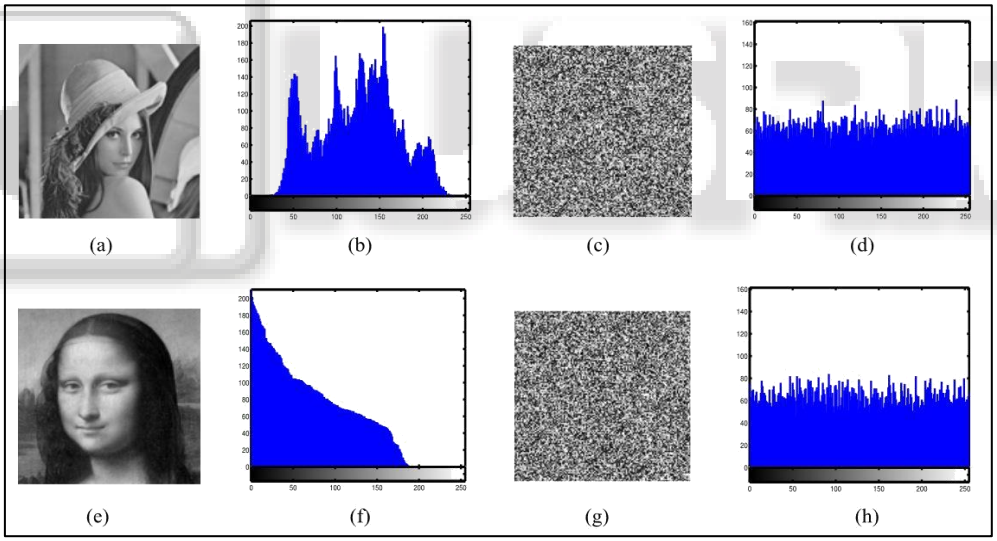

Fig. 2: Histograms of plain and cipher images: (a) Plain image Lena.png. (b) Histogram of Lena.png. (c) Cipher image Lena_enc.png. (d) Histogram of Lena_enc.png. (e) Plain image Mona.gif. (f) Histogram of Mona.gif. (g) Cipher image Mona_enc.gif. (h) Histogram of Mona_enc.gif.

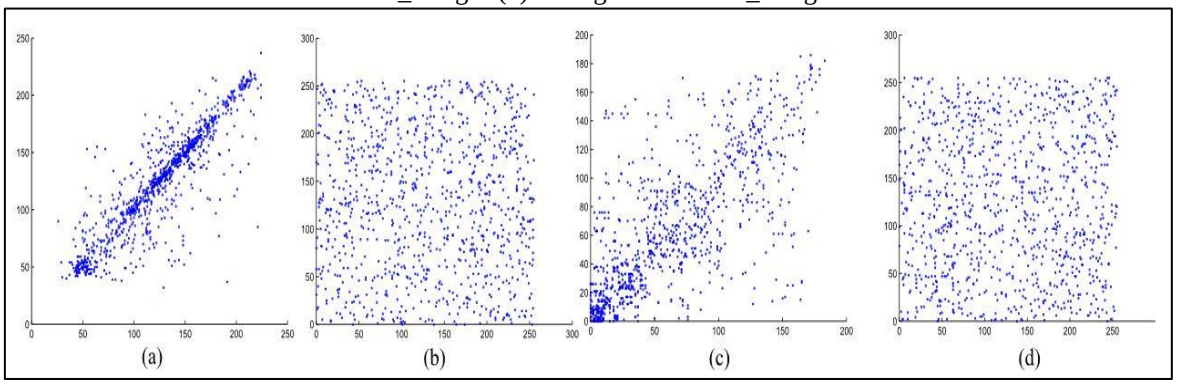

Fig. 3: Correlation of two horizontally adjacent pixels: (a) Plain image (Lena.png). (b) Cipher image (Lena_enc.png). (c) Plain image (Mona.gif). (d) Cipher image (Mona_enc.gif)

2) Correlation of Two Adjacent Pixels

The pixels in an ordinary image are highly correlated with their adjacent pixels either in horizontal, vertical, diagonal or anti-diagonal directions. However, a secure encryption scheme should maintain sufficiently low correlation in the adjacent pixels in the cipher image. To compare the correlations of adjacent pixels, we have calculated the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, two diagonally adjacent pixels, and two antidiagonally adjacent pixels in the plain and cipher images respectively. We have performed the following procedure to find out the correlation between two adjacent pixels. We have randomly selected 1000 pairs of adjacent pixels from the plain image. Then, we have calculated their correlation coefficient using the following formulas:

$$cov(x,y) = E((x - E(x))(y - E(y))) \quad (3)$$

$$f_{xy} = \frac{cov(x,y)}{\sqrt{var(x)}\sqrt{var(y)}} \quad (4)$$

Where x and y are gray-levels of two adjacent pixels in the image. Fig. 3(a) and Fig. 3(c) show the correlations of two horizontally adjacent pixels in the original images in Fig. 2(a) and Fig. 2(e) respectively, whereas Fig. 3(b) and Fig. 3(d) represent the correlation of two horizontally adjacent pixels in corresponding cipher images in Fig. 2(c) and Fig. 2(g) respectively. From the figures, it can be seen that the two horizontally adjacent pixels in the plain images are highly correlated but the correlation between the two adjacent pixels in the cipher images is negligible. Similar results are obtained for the vertical, diagonal, and anti-diagonal directions, and shown in Table I

C. Differential Attack Analysis

To test the impact of a one-pixel change in the original image, two common measures are used: the number of pixels change rate (NPCR) and the unified average changing intensity (UACI). Let C1 and C2 be two cipher images, whose corresponding plain images have only one pixel difference. We define the gray values of the pixels at grid (i, j) in C1 and C2 as C1 (i, j) and C2 (i, j), respectively. We declare a bipolar array D with the same size as image C1 or C2.The value of D(i, j) is determined by C1 (i, j) and C2 (i, j) as follows; if C1 (i, j)=C2 (i, j) then D(i, j) = 1, otherwise D(i, j)=0. The NPCR is defined by the following formula:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (5)$$

Input Images	Horizontal	Vertical	Diagonal	Anti-diagonal
Lena.png	0.8960	0.9473	0.8647	0.8907
Lena_enc.png	0.0027	0.0019	0.0070	0.0034
Mona.gif	0.7798	0.7719	0.7093	0.7343
Mona_enc.gif	0.0197	0.0356	0.0061	0.0280

Table 1: Correlation Coefficients of Adjacent Pixels

Input Images	NPCR	UACI
Lena_enc.png	99.676	33.462
Mona_enc.gif	99.621	33.422

Table 2: NPCR and UACI Test Result

where W and H are the width and height of C1 or C2. Similarly, the UACI is defined by the following equation:

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C1(i,j) - c2(i,j)|}{255}\right] \times 100\% \quad (6)$$

The NPCR calculates the percentage of different pixel numbers between the two images and the UACI calculates the average intensity of differences between the two images. To calculate the NPCR and UACI value for our proposed algorithm, we have changed one pixel value of the two plain images (Fig. 2(a) and Fig. 2(e)) and generated the corresponding cipher images. The results of the NPCR and UACI tests are shown in Table II. From the results, it can be seen that our proposed encryption scheme passes both the NPCR and UACI randomness tests.

D. Information Entropy Analysis

Information entropy presents the degrees of uncertainty in the system and can be used to analyse the indeterminateness of an encryption scheme. We can formally define the entropy, H(m) of any message m as follows

$$H(m) = \sum_{i=1}^{N} P(mi) \times \log_2 \frac{1}{P(mi)} \quad (7)$$

where $p(m_i)$ denotes the probability of the symbol m_i and N is the total number of symbols. If the output of a cipher emits 2^n symbols, then the entropy should be n. As an example, the ideal entropy of a 256-gray scale image must be 8 since the pixel elements have 2^8 possible values. If the entropy value is less than 8, there exists a certain degree of predictability, which threatens the security of the encryption algorithm.

Input Images	Plain image	Cipher image
Lena	7.4102	7.9988
Mona	7.3499	7.9884

Table 3: Information Entropy of Plain and Cipher Images

The Table III shows the entropies for both plain images and cipher images. It can be seen that the entropy values of cipher images are much closer to the expected value of 8. This means that the chance of information leakage is negligible and the proposed scheme is secure against entropy attack.

E. Known Plaintext and Ciphertext Only Attack

The known plaintext attack (KPA) is an attack based on having samples of both plaintext and corresponding ciphertext. Now, using this information, the attacker tries to find the secret key used in encryption and decryption process. However, KPA is not feasible in this encryption scheme since different plain images are encrypted using a different key stream. Hence, it is not possible to obtain useful information by encrypting any special image because the resultant cipher depends upon a random number of operations on the basis of the key stream.

The ciphertext only attack (COA) is an attack model used in cryptanalysis when the attacker has access only to a set of ciphertext. For a given set of ciphertext, if the attacker can determine corresponding plaintext then COA is successful. Suppose an adversary performs exhaustive search on the first 1024 bits of a cipher image to

retrieve one-sixteenth segment of the plain image. The possible combinations will then be a number of $2^{1024} \approx 1.797 \times 10^{308}$ which indicates that the COA attack on this proposed encryption scheme is infeasible.

VI. PERFORMANCE COMPARISON

\We implemented our proposed encryption scheme in MICA2 sensor mote, consists of a microprocessor (ATmega128L) operating at 7.3728 MHz, 128 KB program memory and 4 KB data memory [29]. The mote supports an event driven operating system known as TinyOS [30] and a high level programming language based on components called nesC. Moreover, we have evaluated our crypto-system in ATEMU emulator to perform high fidelity large scale sensor network emulation studies in a controlled environment. Here, we present the

Algorithm	CPU Cycles	Time (ms)	RAM (bytes)	ROM (bytes)
SkipJack	91224	12.353	292	7218
AES	68512	9.287	324	6994
LED	589652	78.972	378	5970
TWINE	128896	17.477	384	5280
BCC	91286	12.547	976	6240
Proposed	62396	8.547	542	5326

Table 4: Comparing CPU and Memory Usage

results of our experiments performed using both simulator TOSSIM and emulator ATEMU [31]. NIST recommended 128-bit elliptic curve domain parameters over prime field were used in our experiments to generate the key pool in the key establishment phase. However, the computational cost of key establishment phase is not considered in our experiments since this process is done only once during the setup phase. AES, non-optimized SkipJack, LED, TWINE and BCC protocols are also implemented in TinyOS environment and the results are compared with our proposed cryptographic scheme as shown in Table IV.

Operation speed indicates time complexity and is an important factor for performance evaluation. We have used ATEMU to get the total CPU cycles required to get the total CPU cycles required to encrypt 32 bytes data for MICA2 sensor mote. On the other hand, using TOSSIM, we have calculated the memory and total encryption time in milliseconds for SkipJack, AES, LED, TWINE, BCC and our proposed cipher. The results in the table indicate that our proposed algorithm performs better in terms of CPU elapsed time (8.547 ms) using only 62396 CPU cycles. For AES, SkipJack and BCC, the number of CPU cycles and encryption time is higher compared to our scheme, while the elapsed time and required CPU cycles are almost double for TWINE and nine times higher for LED protocol. In case of memory consumption, it can be seen that TWINE is more efficient than the other protocols. Although our algorithm uses marginally more memory compared to TWINE, it is less than that of SkipJack, AES, LED and BCC. Overall the proposed algorithm performed significantly better than other algorithms.

VII. CONCLUSION

This paper presents a fast, provably secure and robust block cipher for WSN applications. The cryptographic scheme incorporates the benefits of elliptic curve operations, chaotic map and genetic cryptography to provide data confidentiality. The proposed encryption scheme randomly selects different secret keys (x_i, y_i) rather than fixed parameters for every session. This mechanism makes the crypto-system harder to break for adversaries. Another advantage of the proposed cryptographic scheme is the ability to encrypt both text and image data. The scheme is thus more suitable for use in multimode sensors. Theoretical analyses and experimental results show that the proposed block cipher is provably secure and is more resource efficient in terms of resource consumption. However, the encryption scheme has a few limitations: i) since the proposed algorithm uses blocks of plaintext (i.e., binary codes), it requires padding when the size of plaintext is smaller than the pre-defined block size; ii) the initial parameters must be pre-distributed using a secure channel or a key exchange mechanism. In our future work, we will implement the protocol for audio and video encryption. Moreover, we plan to implement it in large scale sensor networks to evaluate overall message throughput and latency.

REFERENCES

[1] G. R. Sakthidharan and S. Chitra, "Asurvey on wireless sensor network:An application perspective," in Proc. ICCCI , Jan. 2012, pp. 1–5.
[2] M. Cazorla, K. Marquet, and M. Minier, "Survey and benchmarkof lightweight block ciphers for WSNs," inProc. Int. Conf. Secur.Cryptograph., Jul. 2013, pp. 543–548.
[3] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer securityarchitecture for wireless sensor networks," in Proc. 2nd Int. Conf.SenSys, 2004, pp. 162–175.
[4] Intel Architecture Software Developer's Manual , Intel Corporation, Santa Clara, CA, USA, 1997.
[5] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," J. Cryptol. , vol. 18, no. 4, pp. 291–311, 2005.
[6] J. Kelsey, B. Schneier, and D. Wagner, "Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," in Proc. 1st Int. Conf. ICICS, 1997, pp. 233–246.
[7] E. Yarrkov.(2010). Cryptanalysis of XXTEA. [Online].Available:http://eprint.iacr.org/2010/254.pdf
[8] Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique crypt- analysis of the full AES," in Advances in Cryptology (Lecture Notes in Computer Science), vol. 7073. Berlin , Germany: Springer-Verlag, 2011, pp. 344–371.
[9] T. Xiao-Jun, W. Zhu, and Z. Ke, "A novel block encryption scheme based on chaos and an S-box for wireless sensor networks," J. Chin. Phys. B , vol. 21, no. 2, p. 020506, 2012.
[10] De Cannière, O. Dunkelman, and M. Kneževi´ c, "KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers," in

Cryptographic Hardware and Embedded Systems , vol. 5747. Berlin, Germany: Springer-Verlag, 2009, pp. 272–288.
[11] S. Chen, X. Zhong, and Z. Wu, "Chaos block cipher for wireless sensor network," J. Sci. China Ser. F, Inf. Sci., vol. 51, no. 8, pp. 1055–1063, 2008.
[12] J. Yang, D. Xiao, and T. Xiang, "Cryptanalysis of a chaos block cipher for wireless sensor network," Commun. Nonlinear Sci. Numer. Simul.,vol. 16, no. 2, pp. 844–850, 2011.
[13] Y. Liu and S. Tian, "Design and statistical analysis of a new chaos block cipher for WSN," in Proc. IEEE ICITIS , Dec. 2010, pp. 327–330.

