Dynamic adjustment of the chaos-based security in real-time energy harvesting sensors

Conference Paper · August 2013

CITATION
1

READS
397

1 author:

Mousa Farajallah
Palestine Polytechnic University
24 PUBLICATIONS **98** CITATIONS

SEE PROFILE

2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing

Dynamic Adjustment of the Chaos-based Security in Real-time Energy Harvesting Sensors

M. Farajallah, S. El Assad, Member IEEE
IETR
University of Nantes
Nantes, France

Safwan.elassad@univ-nantes.fr;
mousa.farajallah@etu.univ-nantes.fr

Maryline Chetto, Member IEEE
IRCCyN
University of Nantes
Nantes, France

maryline.chetto@univ-nantes.fr

***Abstract*— Wireless Sensor Networks (WSNs) are a growing field of research since they are used in many applications. Nevertheless, they are subject to many requirements such as real-time constraints, energy limitations, and security requirements for the communications. Energy Harvesting is a new paradigm in WSNs: sensor nodes are powered by energy harvested from the ambient, rather than by non-rechargeable batteries which permits a potentially perpetual operation. However, energy harvesting poses new challenges in the design of WSNs, in that energy availability fluctuates over the time. In this paper, we investigate the following fundamental question: how should the harvested energy be managed to guarantee data security in all circumstances?**

Our contributions in this paper are twofold: First, we propose the Deadline Mechanism to dynamically cope with energy shortage and guarantee the highest possible quality of protection by the use of different encryption algorithms. Second, we design a new chaos based cryptosystem suitable for WSNs. Indeed, the proposed encryption/decryption scheme is robust against all known attacks. Experiments show that it is at least 7 times faster than the AES algorithm and also, faster than many chaos based cryptosystems of the literature. Our objective is to identify low-complexity policies that achieve close-to-optimal performance, in terms of maximizing the average long-term importance of the reported data.

Keywords-Real-time; scheduling; Quality of Service; reliability; information security; secure transactions; chaos based crypto-system.

I. Introduction

A wireless sensor network consists of sensor nodes deployed over a geographical area for monitoring physical phenomena like temperature, humidity, vibrations, seismic events, and sensitive information such as enemy movement on the battlefield or the location of persons in a building. Typically, a sensor node is a tiny device that includes four basic components: a sensing subsystem for data acquisition from the physical surrounding environment, a processing subsystem for local data processing and storage, a wireless communication subsystem for data transmission, and a power source to supply the energy needed by all the subsystems to perform the programmed tasks. The power source often consists of a battery with limited life time. In many applications, it is impossible or costly to recharge the battery because nodes may be deployed in hostile or unpractical environments. The sensor network should have a lifetime long enough (may be years) to fulfill the application requirements. Therefore, the question is "how to prolong the network lifetime to such a long time" [1].

Sensors communicate broadcasting messages to each other which makes them vulnerable to different types of attack, such as eavesdropping, malicious modification on insertion of unauthorized data in packages [2]. To prevent attacks, we should implement security services in WSNs. Among them, the most important are confidentiality, authentication of nodes as well as data integrity. Most of traditional security mechanisms are not sufficiently efficient.

All existing algorithms including cryptographic algorithms adapted to WSNs aim to reduce the energy consumed and thus to maximize the lifetime of the networks. Nevertheless, in the paradigm of energy harvesting sensor networks, the computing/communication system needs to be energy-neutral i.e. to consume as much energy as harvested. In this context, new algorithms have to be developed for power management, routing and other networking issues.

In this paper, we address the question of security in new embedded systems which are supplied by ambient energy. A variety of techniques are available for energy harvesting, including solar and wind powers, piezoelectricity, thermoelectricity, and physical motions. Energy harvesting is perfectly convenient for wireless electronic devices that otherwise rely on battery power. In the energy harvesting paradigm, the lifetime of a network can be considered as infinite. A central question that remains open is about the possibility of fully secure communications with energy harvested sensors. This paper addresses this key question.

We propose an on-line strategy based on software redundancy which, at any time permits to select the adequate encryption algorithm so as to optimize the resulting Quality of Service measured in terms of security and rapidity. Further, we propose a security-aware scheduling strategy which

978-0-7695-5046-6/13 $26.00 © 2013 IEEE
DOI 10.1109/GreenCom-iThings-CPSCom.2013.65

incorporates the Earliest Deadline First (EDF) scheduling algorithm.

The rest of the paper is organized as follows: Section II briefly reviews real-time systems and the associated scheduling algorithms. In Section III, we first describe the Deadline Mechanism as a solution to execute real-time periodic tasks in charge of secured transactions within a hard deadline under energy limitations. Section IV presents the structure details of proposed chaos based cryptosystem. In section V, we present the experimental comparative results obtained by the proposed cryptosystem and the AES algorithm. The paper concludes with Section VI.

II. REAL-TIME COMPUTING

A. *Issues in conventional real-time computing systems*

Data sensing and retrieval in wireless sensor systems have a widespread application. The software on a node generally comprises a set of periodic tasks that consist of streams of jobs. Task periods are usually set by the application requirements. Every job is characterized by a release time, an execution requirement and a deadline. The goodness of such real-time system depends on whether all the jobs of all the tasks can be guaranteed to complete their executions before deadlines. If they can, then we say the task set is feasible.

It has been established for about forty years that the EDF (Earliest Deadline First) scheduling policy which assigns priorities according to urgency is optimal [3]. In case timing constraints cannot be met, one way is to trade computation quality for timeliness. This is often achieved with software redundancy in order to provide a graceful degraded mode. Indeed, a real-time system should continue to operate even in the presence of time starvation or energy starvation with a lower but acceptable Quality of Service.

B. *The Deadline Mechanism*

The so-called *Deadline Mechanism* provides software redundancy in hard real-time periodic task systems [4]. Each task has two versions: *primary* and *alternate* (also called back-up). The primary version contains more functions and produces good quality results, but requires high computation time (and high energy requirement). The alternate version contains only the minimum required functions and produces less precise results with minimum execution requirements. However, if the primary of a task fails (due to time or energy starvation), the execution of the associated alternate should be guaranteed before deadline. The challenge in the implementation of the Deadline Mechanism is consequently twofold: 1) how to guarantee that either the primary or the alternate version of each task be completed in time and 2) how to complete as many primaries as possible to optimize the quality of service.

C. *Scheduling framework*

In [5] the so-called Last Chance strategy is proposed to maximize the number of successful primaries. An offline scheduler reserves time intervals for the alternates. Each interval is chosen so that any alternate starts execution at the latest possible time. At runtime, the primaries are processed during the remaining intervals before their respective alternate. Alternates can preempt any primary when a time interval reserved for the alternates overlaps the execution interval of primaries. Whenever a primary completes successfully, the execution of its corresponding alternate is no longer needed. Hence, the online scheduling algorithm dynamically rearranges the alternate schedule so as to increase processor time available for the execution of primaries (see Figure 2).

III. SECURITY IN ENERGY HARVESTING SYSTEMS

There have been a lot of researches about energy harvesting computing systems from about twenty years. Most of papers address the general problem of task scheduling and dynamic power management so as to guarantee the energy neutral behavior of these systems. In the other side, a lot of works address chaos-based security systems disregarding energy harvesting constraints. To our knowledge, the framework that is proposed in this paper is the first one which deals with chaos-based cryptosystems implemented in an energy harvesting computing/communicating system such as WSNs.

A. *System Model*

Our system consists of four components: the energy source, the energy harvester, the energy storage and the sensor node see Figure 1. The ambient energy source is characterized by an instantaneous charging rate P_r (t) that incorporates all losses. We assume that energy production times can overlap with the consumption times. The energy produced by the source is not considered as controllable and not necessarily predictable. Our system uses an ideal energy storage unit that has a nominal capacity, namely C. Let define E (t) as the residual capacity of the storage unit at time t.

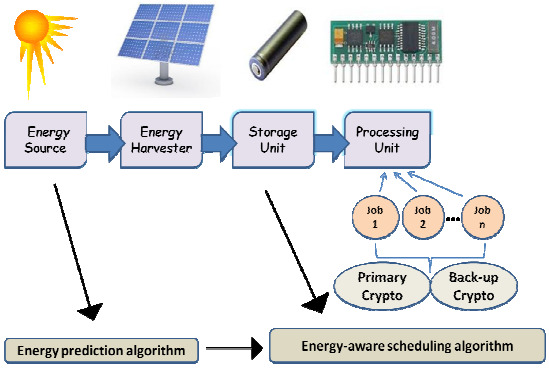


Figure 1. Framework of the energy harvesting system

We consider a uniprocessor sensor node that consumes negligible energy in idle state. Transactions require to be performed periodically. We assume that two encryption algorithms are available. The first one has a greater execution time and greater energy requirement but provides the highest quality of security. The second one is characterized by a very low execution time and energy requirement, called by the alternate in energy starvation situations. Consequently, every

encryption task τ_i is modeled by a five-tuple (Cp_i, Ca_i, Ep_i, Ea_i, T_i) where it is denoted:

- Cp_i respectively Ca_i: the Worst Case Execution Time (WCET) of the primary respectively the alternate of τ_i with $Cp_i > Ca_i$,
- Ep_i respectively Ea_i the Worst Case Energy Consumption (WCEC) of the primary respectively the alternate of τ_i with $Ep_i > Ea_i$,
- T_i the period of τ_i.

We assume that E_i is not necessarily proportional to C_i. Moreover the average incoming power is higher than or equal to the average power consumed by the alternates. And the processor utilization rate due to the alternates is less than one.

B. The scheduling issue

The introduction of energy harvesting capabilities in sensor networks has introduced additional design questions. How to intelligently use the ambient incoming energy to optimize the QoS of the system measured in terms of security? Furthermore, how to adapt the processing activity so as to subsist perennially on a given energy source?

We need a scheduling algorithm which (i) guarantees either the primary or alternate version of each encryption task to be completed in time and (ii) attempts to complete as many primaries as possible. Our basic strategy uses Earliest Deadline as late as possible to pre-allocate time intervals to the alternates. Even if EDF is not the optimal scheduler in energy harvesting systems, we proved recently that EDF is the optimal non idling scheduler [6]. In contrast to an optimal scheduler, EDF is no clairvoyant and consequently easy to implement. Here, we want to delay execution of alternates as much as possible to save both time and energy. At run-time, it attempts to execute primaries first. An alternate will be executed only (1) if its primary fails due to lack of time, lack of energy or manifestation of bugs, or (2) when the latest time to start execution of the alternate without missing the corresponding task deadline is reached. This algorithm has been shown to be effective and easy to implement.

Figure 2, depicts a very simple example where two periodic tasks of respective periods 3 and 9 have to execute before deadline. The Last chance strategy attempts to execute the primaries first. From time 0 up to time 3, primaries execute timely because of sufficient energy available in the storage unit. When primary τ_1 succeeds at 2, the interval reserved for the alternate is removed and additional time could be used for primaries. Here, we do not illustrate the process for reserving energy that should guarantee feasibility of all alternates. Second primary τ_1 fails before completion due to depletion of the storage unit. The primary is aborted and the storage starts recharging since we let the processor inactive until the storage fully replenishes. That permits primary τ_2 and alternate τ_1 to complete successively as third primary τ_1.

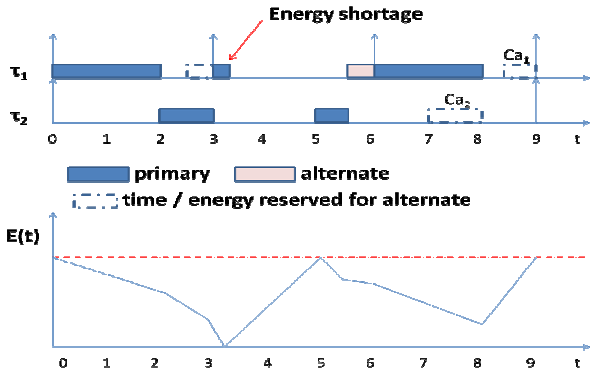


Figure 2. The Deadline Mechanism

In the next section, we propose two chaos-based cryptosystems. The first one is implemented as the primary version that provides the best quality of security (see figures 3 and 4). The second one (the alternate version) is a simplified version of the primary algorithm in which the permutation process has been removed. So, it has lower execution time and lower consumption energy.

IV. ARCHITICTURE OF THE PROPOSED CHAOS-BASED CRYPTOSYSTM

Chaos based applications become more and more widely, since chaos systems have extreme sensitivity for initial parameters. The idea of using chaos for image encryption has become a hot researching topic during the last two decades, but the basic of this idea is too old and it was founded in Shannon's paper [7]. A large number of chaos based cryptosystems for image encryption are introduced during the last decade, most of them are secure but not fast, while the other are fast but not secure, [8-11]. A chaos-based cryptosystem must achieve the security requirements (diffusion and confusion effects, plaintext and key sensitivity attacks, correlation and histogram…), and also to be very speed in order to use it for real time applications.

In this paper we proposed a new chaos based cryptosystem for real time applications. The speed of this cryptosystem is, from our knowledge, better than any cryptosystem has been published, and its level of security is high (from the analysis results).

The proposed cryptosystem consists of four components, a substitution layer, a diffusion layer, a permutation layer, and a chaotic generator. The structure of the cryptosystem is shown in Figure 3, for the encryption part and in Figure 4, for the decryption part. The plain image is divided into *N* blocks, with *N=image size/block size*, where the *block size* value is chosen to be a square value (here 256). The *image size* must be a multiple of the *block size*, so, in case of that *image size* is not a multiple of the *block size*, we pad the last block by *(block size - mod (image size, block size))* bytes.

In encryption part (see Figure 3), the divided blocks are ciphered one by one by applying the substitution layer on the

block for *rs* times to achieve the confusion effect. Next the diffusion layer is used to spread a single byte effect to the other bytes in the same block this layer is repeated *rd* times. Then the output of this layer is forwarded to the permutation layer to exchange the byte position inside the same block to add more confusion effect, permutation layer also is repeated *rp* times. These three chaotic layers are repeated *r* rounds to get the required security level. For each round *r*, new necessary dynamic keys are produced by a robust chaotic generator [12], and then used in each layer.

The decryption process of the proposed cryptosystem is just the reverse operations of the encryption one (see Figure 4). The whole process is almost the same, but the dynamic keys are used in reverse order, and the permutation layer works in reverse order for each block. The counters *rp*, *rd*, and *rs* are used in reverse order.

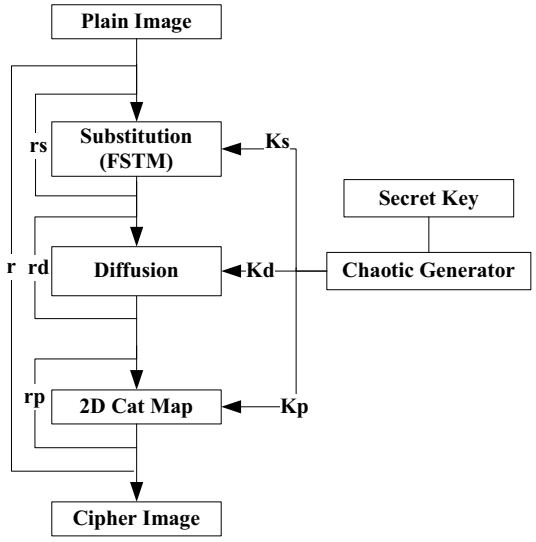


Figure 3. Encryption part of the proposed cryptosystem

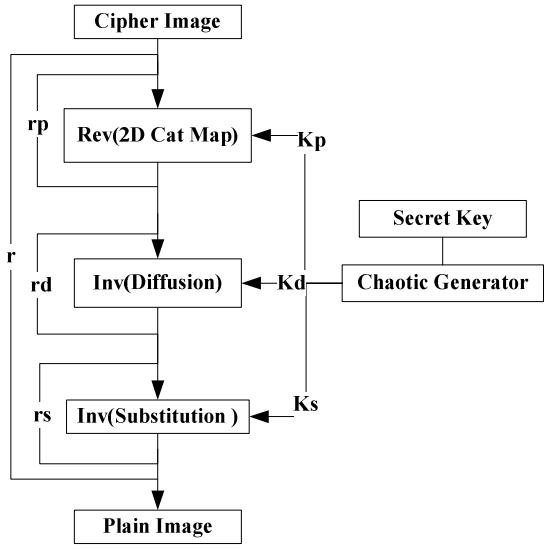


Figure 4. Decryption part of the proposed cryptosystem

In the following three sub sections we present in details the mathematical models of each layer.

A. Substitution layer

The substitution layer is based on the Finite Skew Tent Map (FSTM) that was proposed by Masuda et al [13][14]. We have implemented and evaluated this map as a substitution layer. It changes the value of the current pixel into a new value in accordance with a dynamic key produced by the proposed chaotic generator in each new substitution round. Equation (1), achieves the substitution operation:

$$Y = S_a(X) = \begin{cases} \left\lceil \frac{Q}{a} \times X \right\rceil & ,0 \le X \le a \\ \left\lfloor \frac{Q}{Q-a} \times (Q-X) \right\rfloor + 1 & ,a \prec X \prec Q \end{cases} \quad (1)$$

The inverse substitution of the FSTM is realized by the following equation:

$$Y = S_a(X) = \begin{cases} \xi_1 & ,\theta(Y) = Y and \frac{\xi_1}{a} \succ \frac{Q-\xi_2}{Q-a} \\ \xi_2 & ,\theta(Y) = Y and \frac{\xi_1}{a} \le \frac{Q-\xi_2}{Q-a} \\ \xi_1 & ,\theta(Y) = Y+1 \end{cases} \quad (2)$$

Where

$$\xi_1 = \left\lfloor \frac{a}{Q} \times Y \right\rfloor \quad (3)$$

$$\xi_2 = \left\lceil \left(\frac{a}{Q} - 1 \right) \times Y + Q \right\rceil \quad (4)$$

$$\xi_3 = \left\lceil \frac{a}{Q} \times Y \right\rceil \quad (5)$$

$$\theta(Y) = Y + \xi_1 - \xi_3 + 1 \quad (6)$$

With $Q = 256$.

During the process of substitution we mentioned that the substitution layer needs a dynamic key, the structure of the dynamic keys are:

$$K_s = \left[K_{s_0} \| K_{s_1} \| K_{s_2} \| .. K_{s_{r-1}} \right] \quad (7)$$

$$K_{s_j} = a_j \quad (8)$$

The range of all possible pixel value is limited to [0-255] and the dynamic keys value is limited to [1-255]. So, equations (1) and (2) are implemented as a lookup table for fast execution.

B. Diffusion layer

Diffusion process is needed to make the relationship between the plain and the corresponding ciphered image as complex as possible [7]. The previous substitution layer is not sufficient for diffusion effect, for that and to transfer the

diffusion effect of each byte we introduce a new simple and fast diffusion layer according to the following equation:

$$X_i = Mod((X_i + X_{i-1}),256) \equiv (X_i + X_{i-1}) \ AND \ 255 \quad i = 0,1,2,3...blocksize - 1 \tag{9}$$

The first byte ($i = 0$), X_{-1} is equal to an initial value (a dynamic key value) produced by the chaotic generator.

The inverse diffusion layer is almost the same as equation (9), but with replacing the plus by minus operator as:

$$X_i = Mod((X_i - X_{i-1}),256) = (X_i - X_{i-1}) \ AND \ 255 \quad i = 0,1,2,3...blocksize - 1 \tag{10}$$

Equations (9) and (10) are used to implement the diffusion and the inverse diffusion layers and they are high speed. We use a mathematical rule to replace the slow *Mod* operation by the fast *AND* operation depends on the following rule:

$$Mod(A,B) = A \ AND \ B-1 \quad if \quad B = 2^k \quad k = 1,2..... \tag{11}$$

K is any integer value; in our proposed diffusion layer we chose it to be 8.

C. Permutation layer

The permutation layer in the proposed cryptosystem is implemented using Arnold cat mapping [15]. It consists of matrix multiplication and modulo operation to map the new pixel position from the old position in the same block. This map was tested and analyzed [16][17]. From this fact we introduce a permutation layer based on a modified Arnold cat mapping (standard Arnold map has a weakness of fixed point, to solve this, the parameters r_i and r_j are added to the standard model).

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = Mod\left(\begin{bmatrix} 1 & u \\ v & 1+u\times v \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} r_i + r_j \\ r_j \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right) \tag{12}$$

As the 2D cat map is one-to-one function, which means every point of the square matrix can be transferred to exactly one unique point, so, instead of exchanges the values at position (i_n, j_n) and position (i, j), we use a transfer operation because of its speed compared to the exchange operation that usually used. The block size is M^2, the system parameters u, v, r_i and r_j are in the range of $[0, M-1]$.

The structure of the dynamic keys produced by the chaotic generator during the permutation process is:

$$K_p = \left[K_{p_0} \| K_{p_1} \| K_{p_2} \| ..K_{p_{r-1}} \right] \tag{13}$$

$$K_{p_j} = \left[u_j \| v_j \| rl_j \| rc_j \right] \tag{14}$$

The modulo operation of the equation (12) makes it a non invertible equation but a reversible one. Then, in the decryption part of the proposed cryptosystem, the reverse permutation is achieved by the same equation (12) but it is used in reverse order as well as the dynamic keys of permutation.

V. Performance : Time and Security Evaluation

The proposed cryptosystem was applied to three different images (Cameraman, Peppers, and Parrots) of various sizes. The results listed in the next three subsections (time evaluation, key sensitivity and plain text sensitivity attacks), are measured on the Cameraman of two sizes ($128\times128\times3$ and $512\times512\times3$).

A. Time analysis

To compare our proposed cryptosystem, we executed it and the AES algorithm on the same C compiler on the same machine that has the following characteristics:

1- HP Compaq 8200.
2- Intel® processor Core ™ i3-2100 CPU @ 3.1 GHz.
3- Windows 7.
4- 4 GB RAM

Our proposed cryptosystem reaches the security requirements (Hamming Distance, NPCR, UACI, etc…) from the first round for large size images ($r = 1, rp = 1, rs = 1, rd = 1$), and it needs only two diffusion rounds ($r = 1, rp = 1, rs = 1, rd = 2$), for a small size images. In Table 1 and Table 2 we give the calculation time of our cryptosystem and the AES algorithm for the Cameraman image of sizes $128\times128\times3$, and ($512\times512\times3$) respectively. As we can see from the obtained results that, our cryptosystem is at least 10 times faster than the AES algorithm.

TABLE 1. ENCRYPTION AND DECRYPTION TIMES IN SECONDS OF THE PROPOSED ALGORITHM VS THE AES FOR 128X128X3 IMAGE SIZE

	Encryption	**Decryption**
Proposed Algorithm	0.0018	0.0019
AES	0.016	0.017

TABLE 2. ENCRYPTION AND DECRYPTION TIMES IN SECONDS OF THE PROPOSED ALGORITHM VS THE AES FOR 512X512X3 IMAGE SIZE

	Encryption	**Decryption**
Proposed Algorithm	0.024	0.025
AES	0.257	0.270

It is clear that the time of our proposed cryptosystem in Table 2 is not 16 multiple of the time in Table 1, this come from the idea of that the diffusion layer is consuming 25% of the total encryption time and 31% of the total decryption time. Also the image load time and initialization program for first image size is not 16 multiple of the second image size.

Remark: we have used the AES algorithm given by the following website:

https://code.google.com/p/rikiglue/source/browse/src/frame/aes.cpp?spec=svn9239a0474d811daae909075568688a46134858c6&r=9239a0474d811daae909075568688a46134858c6

B. Plain text sensitivity analysis

Secure cryptosystem must be resistant to differential, chosen and known plain text attacks. That means one bit change on the plain image produce a completely different cipher image. The difference between the two ciphered images is measured by the Hamming distance formula, where C_1 is the encrypted image from the original plain image, while C_2 is the encrypted image from a modified plain image by one bit change, the both encryption processes are done using the same secret key.

$$d_{Hamming}(C_1,C_2)=\sum_{K=1}^{L\times C\times P\times 8} C_1(K)\oplus C_2(K) \quad (15)$$

Another two security parameters, often used by the researchers to test the plain text sensitivity attacks based on bytes, are: Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) [18]:

$$NPCR(C_1,C_2)=\frac{1}{L\times C\times P}\sum_{K=1}^{L\times C\times P} D(K)\times 100 \quad (16)$$

Where

$$D(K)=\begin{cases}1 & if\ C_1(K)\neq C_2(K)\\ 0 & f\ C_1(K)=C_2(K)\end{cases} \quad (17)$$

$$UACI(C_1,C_2)=\frac{1}{L\times C\times P\times 255}\sum_{K=1}^{L\times C\times P}\left|C_1(K)-C_2(K)\right|\times 100 \quad (18)$$

Table 3, presents the obtained results of the average of three parameters *HD*, *NPCR* and *UACT* of 1000 different secret keys, for the following parameters:

$r=1, rp=1, rs=1, rd=1$ for image size of $512\times512\times3$

$r=1, rp=1, rs=1, rd=2$ for image size of $128\times128\times3$

Table 3. *HD*, *NPCR*, and *UACI* values for the plain text sensitivity attack test

Test/Image size	$128\times128\times3$	$512\times512\times3$
HD	0.499835	0.499917
NPCR	99.606	99.609
UACI	33.472	33.451

It is clear from the results in Table 3 that the proposed cryptosystem has high security level and almost optimal.

C. Key sensitivity analysis

A slight change in the secret key must produce incorrect estimate plain image during the decryption process, and must produce completely different ciphered image during the encryption process. To do this test, we change one bit in the secret key and encrypt the same plain image. The same security parameters of the previous section are used, but in this test all results are measured for one round. Table 4 presents the obtained results of the test.

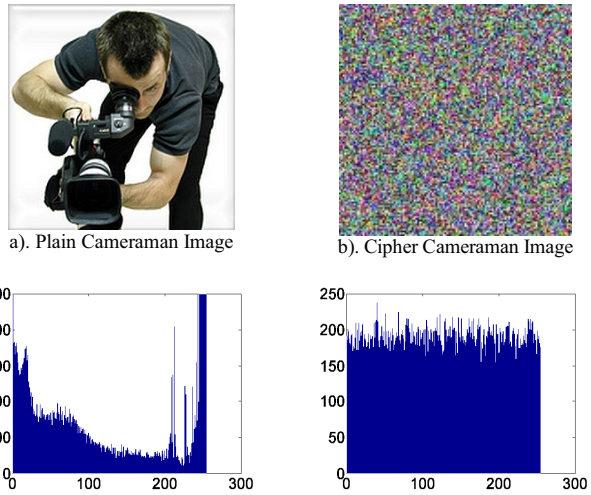
Table 4. *HD*, *NPCR*, and *UACI* values for the key sensitivity attack test

Test/Image size	$128\times128\times3$	$512\times512\times3$
HD	0.500003	0.499989
NPCR	99.609	99.609
UACI	33.463	33.462

The results in Table 4 show that the proposed cryptosystem is highly resistant to the key sensitivity attack.

D. Histogram analysis

We calculated the histogram of different plain images and their corresponding ciphered images. Figure 5, shows the obtained results of the Cameraman image.



a). Plain Cameraman Image b). Cipher Cameraman Image

c). Histogram of the Plain image d). Histogram of the cipher image

Figure 5. Histograms of the plain and its ciphered images

In part c) of Figure 5, we show the histogram of the plain image, and for more clarity, we plot the histogram and limit the maximum value of the y-axis to be 500. Indeed, the histogram values for all colors (except the black and white colors) are less than 500, and we found that the white color is repeated more than 7500 times (16%) and the black color is repeated more than 3300 times (7%). It is clear from part c) that the pixel values have a pattern, while in part d) (histogram of the cipher image) the distribution of the pixel values is almost uniform and significantly different from the histogram of the plain image.

In addition, to ensure the histogram uniformity of the ciphered images (Cameraman, Pepper, and Parrot), we apply the chi-square test given by the following equation:

$$\chi^2_{\exp}\sum_{i=0}^{Ny-1}\frac{(O_i-E_i)^2}{E_i} \quad (19)$$

Table 5, presents the calculated chi-square value for each histogram. As, we can see, the experimental values are less than the theoretical one which is 293 in case of alpha=0.05 and number of intervals = 256.

Table 5. Chi square test results

	Chi-square Value
Cameraman	255.12
Peppers	260.36
Parrots	259.44

E. Correlation analysis

To measure the correlation coefficient, we randomly selected 8000 pairs of adjacent pixels in vertical, horizontal, and diagonal directions from the plain and their ciphered images [11]:

$$\rho_{xy}=\frac{\mathrm{cov}(x,y)}{\sqrt{D(x)\times D(y)}} \quad (20)$$

Where:

$$\mathrm{cov}(x,y)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))\times(y_i-E(y)) \quad (21)$$

$$D(x)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))^2 \quad (22)$$

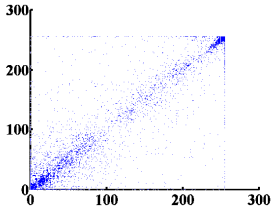
$$E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i \quad (23)$$

x_i and y_i are the gray values of two adjacent pixels in the plain images or in the ciphered images and N is the sample size (8000). The correlation values of the three tested images are listed in Table 6. Figure 6, shows the correlation curves of the adjacent pixels in horizontal, vertical and diagonal directions.

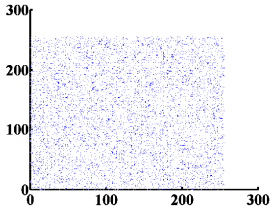
It can be seen from Table 6 and Figure 6 that, the strong and high correlation coefficients in the plain image are transformed to almost negligible correlation coefficients in the cipher image, this means that, the cipher image is secure enough.

TABLE 6. CORRELATION COEFFICIENTS OF PLAIN AND CIPHER IMAGES

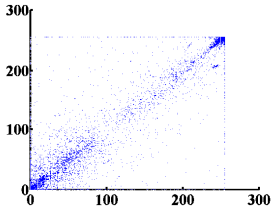
	Plain image	Cipher Image	Image Name
Horizontal	0.898492	0.010523	Cameraman
Vertical	0.925182	0.010650	
Diagonal	0.851377	0.010842	
Horizontal	0.932197	0.009870	Peppers
Vertical	0.940916	0.007380	
Diagonal	0.888588	0.010905	
Horizontal	0.856771	0.013007	Parrots
Vertical	0.877517	0.010761	
Diagonal	0.797259	0.010914	



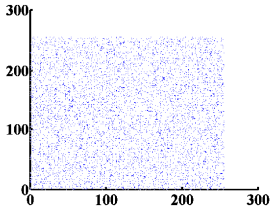
a-) Correlation of adjacent pixels in vertical direction of the plain image



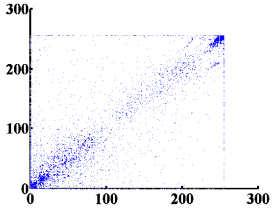
b-) Correlation of adjacent pixels in vertical direction of the cipher image



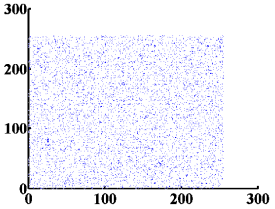
c-) Correlation of adjacent pixels in horizontal direction of the plain image



d-) Correlation of adjacent pixels in horizontal direction of the cipher image



e-) Correlation of adjacent pixels in diagonal direction of the Plain image



f-) Correlation of adjacent pixels in diagonally direction of the cipher image

Figure 6. Correlation analysis of the plain and ciphered images in vertical, horizontal, and diagonal directions

VI. CONCLUSION AND FUTURE WORK

The importance of security support is greatly increasing, because more real-time systems are being networked. To securely transmit sensitive data across the network, cryptographic techniques are used to protect the data. Thus, real-time tasks can employ various encryption algorithms depending on processor time or/and energy availability. New generation embedded systems including WSN use energy harvesting capabilities which introduce new design questions. In this paper, we propose the Deadline Mechanism to provide degraded mode when energy or time do not permit the system to behave normally. In other words, the security is degraded in a controlled manner by commuting on less consuming encryption algorithm which nevertheless guarantees acceptable degradation in terms of security level while meeting all the deadlines. In this paper we designed and tested a chaos-based cryptosystem suitable for WSNs. The obtained results show that the proposed encryption/decryption scheme is very efficient in terms of robustness and of computing time due to the low complexity of the designed structure.

REFERENCES

[1] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, " Energy conservation in wireless sensor networks : a survey, AD Hoc Networks Journal, vol.7, no.3, pp. 537-568, 2009.

[2] Y. Wang, G. Attebury, B Ramamurthy," A survey of security issues in wireless sensor networks", IEEE communications survey and tutorials, 2nd quarter 2006, vol.8, no.2, pp. 2-23, 2006.

[3] C.-L. Liu, J.-W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment". Journal of the Association for Computing Machinery, Volume 20, Issue 1, pp. 46-61, 1973.

[4] H. Chetto, M. Chetto, "Some Results of the Earliest Deadline Scheduling Algorithm", IEEE Transactions on Software Engineering, Vol 15, Issue 10, pp. 1261-1270, 1989.

[5] H. Chetto, M. Chetto, "An adaptive schedulign algorithm for fault-tolerant real-time systems", Software Engineering Journal, Vol 6, issue 3, p. 93 – 100, 1991.

[6] M. Chetto, A. Queudet "A Note on EDF Scheduling for Real-Time Energy Harvesting Systems", IEEE Trans. On Computers, In press, 2013.

[7] Shannon, C, "Communication Theory of Secrecy Systems". Bell System Technical Journal Vol.28, no.4, pp. 656–715, 1949.

[8] D. Socek, S. Li, S.S Magliveras, and B. Furht, "Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption," in First IEEE International Conference on Security and Privacy for Emerging Areas in Communications Networks, Athens, Greece, 2005, pp. 406 - 413.

[9] H. Yang, K.W Wong, X. Liao, W. Zhang, and P. Wei, "A fast image encryption and authentication scheme based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3507–3517, 2010.

[10] G. Chen, Y. Mao, and C.K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," Chaos, Solitons & Fractals, vol. 21, no. 3, pp. 749-761, 2004.

[11] C.Y. Song, Y.L. Qiao, and X.Z. Zhang, "An image encryption scheme based on new spatiotemporal chaos," Optik, October 2012.

[12] S. El Assad (85%), H. Noura (15%), "Generator of chaotic Sequences and corresponding generating system" WO Patent WO/2011/121,218, 2011.

[13] N. Masuda, G. Jakimoski, K. Aihara, and L. Kocarev, "Chaotic block ciphers: from theory to practical algorithms," IEEE Transaction on Circuits and Systems, vol. 53, no. 6, pp. 1341-1352, 2006.

[14] N. Masuda, K. Aihara,"Cryptosystems with discretized chaotic maps", IEEE Transaction on Circuits and Systems, vol. 49, no. 1, pp. 28–40, 2002.

[15] V. I. Arnold and A. Avez, Ergodic Problems in Classical Mechanics Benjamin, New York, 1968

[16] J. FRIDRICH, "Symmetric Ciphers Based no Two-Dimensional Chaotic Maps," International Journal of Bifurcation and Chaos, vol. 8, no. 6, pp. 1259-1284, 1998.

[17] S. El Assad, "Chaos Based Information Hiding and Security," in 7th International Conference for Internet Technology and Secured Transactions, IEEE, London, United Kingdom, 10-12 Dec. 2012, pp. 67-72. Invited paper.

[18] Eli Biham and Adi Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," in Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, California, USA, 1990, pp. 2-21.