How to guarantee Secured Transactions with QoS and Real-Time constraints

Maryline Chetto
IRCCyN
LUNAM University
Nantes, France

Hassan Noura, Safwan El Assad, Mousa Farajallah
IETR
LUNAM University
Nantes, France

Abstract— **A growing number of systems are highly connected and distributed over the internet. These systems require innovative software services and infrastructures in order to guarantee security and reliability. This issue has a particular relation with a wide range of topics such as real-time scheduling and encryption with Quality of Service constraints which are the main interests of this paper. The proposed cryptosystem has acceptable level of security but it is at least 41 times faster than the AES algorithm.**

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

I. Introduction

There exist a growing number of systems that have real-time and security considerations, because sensitive data and processing require special safeguard and protection against unauthorized access. In particular, a variety of motivating real-time applications running on wireless sensors networks require security protections to completely fulfill their security-critical needs.

Each site may have several optional encryption algorithms where each one is assigned a corresponding security level. A high level generally implies the strongest yet slowest encryption function among the alternatives. Computation overhead caused by encryption mainly depends on the cryptographic algorithms used and the size of the data to be protected. An increasing number of applications demand both real-time performance and security. We investigate here the problem of scheduling independent real-time encryption tasks with various security levels.

We propose an on-line and adaptive strategy based on the Deadline Mechanism 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 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, each one in charge of achieving a secured transaction within a hard deadline. Section IV presents typical algorithms for implementing encryption with different QoS requirements. In section V, we present the experimental comparative results obtained by three distinct encryption algorithms.

The paper concludes with Section VI.

II. Real-time scheduling

A. Background materials

A real-time system comprises a set of tasks where each task consists of a stream of jobs also called instances. The task set can be scheduled by a number of policies generally based on priorities and using preemptively in that sense that a high priority task may suspend a lower priority one in execution. The success of a real-time system depends on whether all the jobs of all the tasks can be guaranteed to complete their executions before their deadlines. If they can, then we say the task set is feasible or schedulable.

The issue of scheduling for real-time applications was previously reported in the literature. Conventional real-time scheduling algorithms are fixed priority based such as Rate Monotonic (RM) algorithm, and dynamic priority based such as the optimal one; Earliest Deadline First (EDF) [1].

B. Quality of Service requirements

In case both the timing and computation-quality constraints cannot be met, one way of meeting the timing constraints is to trade computation quality for timeliness. This is often achieved with the use of redundant programs.

III. The Deadline Mechnanism

A. Description

The so-called Deadline Mechanism combines two methods to provide software fault tolerance in hard real-time periodic task systems [2]. Each periodic task has two versions: primary and alternate. The primary version contains more functions and produces good quality results, but its correctness is more difficult to verify. The alternate version contains only the minimum required functions and produces less precise results and its correctness is easy to verify. However, if the primary

This work has been realized in the framework of the GreenEmbedded project which is part of the Cedre PHC (Programme Hubert Curien) between the French Government and the Lebanon Government.

of a task fails (due to manifestation of a bug) during its execution or if its successful completion cannot be guaranteed (due to insufficient processor time), we must activate the alternate of the task.
The challenge in the implementation of the Deadline Mechanism is 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.

B. Scheduling framework

Chetto and Chetto [3] proposed a last chance strategy to maximize the number of primaries scheduled. An offline scheduler reserves time intervals for the alternates. Each such interval is chosen so that any alternate starts its execution at the latest possible time. At runtime, the primaries are scheduled during the remaining intervals before their alternates. The alternates can preempt a primary when a time interval reserved for the alternates is reached. Whenever a primary is completed successfully, the execution of its corresponding alternate is no longer needed and, hence, an online scheduling algorithm must dynamically deallocate the time interval(s) reserved for the alternate so as to increase the processor time available for the execution of other primaries.

Our algorithm is based on the dynamic priority-driven preemptive scheduling scheme, EDF. We first used an offline EDF, called earliest-deadline-first as late as possible (EDL), to reserve time intervals for the alternates. Then, at runtime, we used any online preemptive scheduling algorithm to schedule the primaries and, whenever a primary is successfully completed the reserved time intervals for the alternates are reconstructed/modified. Such reconstruction is achieved by removing the alternate corresponding to the completed primary and rescheduling the remaining alternates according to EDL from the time the corresponding primary was completed to the end of the current planning cycle. The reconstruction takes a significant amount of time, making the online overhead of their algorithm high.

IV. ENCRYPTION

A. General description of the proposed encryption scheme

We suggest the use of three Encryption/Decryption algorithms for secure data transaction. Two of them are very well known from the literature. It is a question of the advanced encryption standard algorithm (AES) [12] and the Enhanced 1-D Chaotic Key Based Algorithm for Image Encryption (ECKBA) [10]. The third one that we propose here, concerns a very fast and almost robust scheme for image encryption including a strong integer chaotic generator that produces robust pseudo chaotic integer numbers [4-13]. Hence, the proposed crypto-system is built using an initialization layer followed by an scrambling procedure of bytes based on 2D-cat map. Thus, the advantage of this algorithm is its simplicity and that it achieves permutation and substitution value of the original image very fast. Consequently, it is suitable for software and hardware implementation. Also the keystream (parameters of cat map) is generated by a new scheme of chaotic generator with large key and a 32-bits finite precision in integer representation to facilitate and accelerate hardware implementation.

The theoretical and experimental results indicate that this new scheme is almost efficient, and very faster than the AES and the ECKBA algorithms. We note that the proposed encryption changes the statistical characteristics of the original image to random image. So, it is very difficult for an unauthorized person to break this crypto-system. Also, it is immune to statistical, differential, chosen/known-plaintext attacks, and brute force attack.

In figure 1, we give the principle of the encryption scheme.

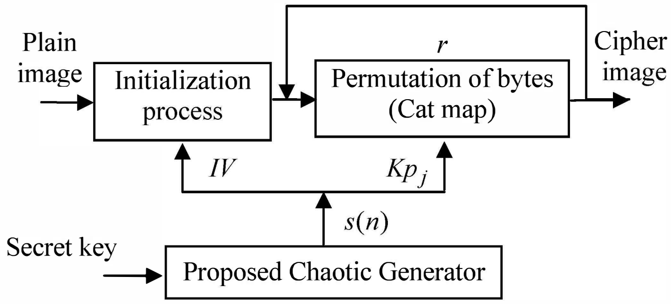

Fig. 1: Principle of the proposed encryption scheme

In contrast with the most recent algorithms, we show that an uncomplicated process of chaotic permutation network (PN) with variable control parameters can form a robust chaotic cryptosystem, as seen in figure 1. This is the main idea of the present paper.

The process of proposed encryption scheme is running as follow:

First, we read the image and express it with a decimal matrix I of size $L\times C\times P$, where L is the number of lines, C is the number of columns, and P is the number of planes. Next, we transform the plain image I of gray value (bytes) into 1-D line IL with length $L\times C\times P$. Then, we apply one time the initialization process on IB, as described in *IV-B*. After that, and for each iteration, we apply the process of permutation using 2-D cat map, as described in *IV-C*.

B. Initialization process

This process protects the data from chosen/known-plaintext attacks. To apply the process of initialization, the IL image is first divided on nb blocks, each of size N=32 bits = 4 bytes.

$$IB = \{B_1, B_2, \cdots, B_k, \cdots, B_{nb}\}$$

With:

$$nb = L \times C \times P / 4$$

Then, the output of this process is given by:

$$\begin{cases} BI_k = B_k \oplus IV \; if \; \mathrm{mod}(k,2) \neq 0 \\ BI_k = \mathrm{mod}\left(B_k + IV, 2^N\right) if \; \mathrm{mod}(k,2) = 0 \end{cases} \qquad (1)$$
$$k = 1, 2, \cdots, nb$$

where *IV* is an initialization vector, produced by the chaotic generator.

Before applying the permutation process, we express the output of the initialization process into 1-D line image in bytes, *IL*.

C. Permutaion process based on cat map

The permutation process, based on the cat map is calculated in a very efficient manner, as compared to the basic calculation.

$$\begin{bmatrix} Ml' \\ Mc' \end{bmatrix} = Mod\left(\begin{bmatrix} 1 & u \\ v & 1+u\times v \end{bmatrix} \times \begin{bmatrix} Ml \\ Mc \end{bmatrix} + \begin{bmatrix} rl+rc \\ rc \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right) \qquad (2)$$

Where Ml *and* Mc are square matrixes, given by the following form:

$$Ml = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 2 & 2 & & 2 \\ \vdots & & \ddots & \vdots \\ M & M & \cdots & M \end{pmatrix}; \; Mc = \begin{pmatrix} 1 & 2 & \dots & M \\ 1 & 2 & & M \\ \vdots & & \ddots & \vdots \\ 1 & 2 & \cdots & M \end{pmatrix}$$

And *Ml¢, Mc¢* are the permuted byte positions of the Ml *and* Mc matrixes. The image size M is equal to $M = \left\lfloor \sqrt{IL} \right\rfloor$ and must of the form $M = 2^q$ with q even. $\lfloor x \rfloor$ represents the integer part of x (floor of x).

The structure of the dynamic permutation key is

$$Kp = \left[Kp_1 \| Kp_2 \| \cdots \| Kp_r \right]$$

with

$$Kp_j = \left[u_j, v_j, rl_j, rc_j \right] \quad j = 1, \cdots, r$$

The parameters *u*, *v*, *rl*, *rc* are selected as follows: $0 \le u, v, rl, rc \le M - 1 = 2^q - 1$, where $q = \log_2(M)$ is the necessary number of bits to represent each of the four parameters. If *q* is odd, then after each iteration, the last bytes with length equal to ($L \times C \times P - M^2$) are swapped and put in the beginning of the matrix to be permuted.

The inverse permutation process is similar to the permutation one. The only difference is that the last index *(M, M)* of the permuted matrix will be the start point and the order of the generated keys must be the reverse.

D. Proposed chaotic generator

The proposed chaotic generator consists in two perturbed discretized chaotic maps (PWLCM and Skew tent) connected in parallel as seen in figure 2 [4]. The two generated chaotic sequences s_1 and s_2 are then bitwise exclusive-or to give the dynamic *IV* vector and the dynamic keystream $Kp_j, j = 1, \cdots, r$.

In order to extend the periodicity and to control it, we apply a perturbation technique as in (El Assad, 2010).
The disturbance is applied using the last *k* bits of the linear feedback shift registers (*LFSR*) outputs.

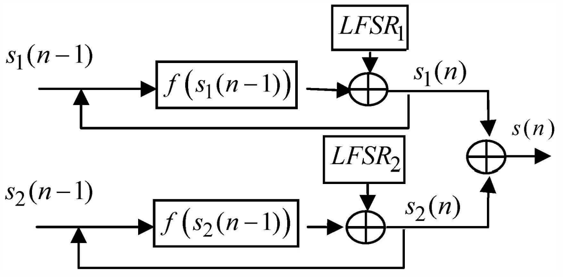

Fig. 2. Proposed chaotic generator

The discrete equations of the PWLCM and the Skew tent maps are given bellow :

$$s_1(n) = NLF_1[u_1(n-1), p_1]$$
$$= \begin{cases} \left\lfloor 2^N \times \dfrac{u_1(n-1)}{p_1} \right\rfloor & if \;\; 0 \le u_1(n-1) < p_1 \\ \left\lfloor 2^N \times \dfrac{2^N - u_1(n-1)}{2^N - p_1} \right\rfloor & if \;\; p_1 \le u_1(n-1) < 2^{N-1} \\ NLF_1[2^N - u_1(n-1)] & Otherwhise \end{cases} \qquad (3)$$

$$s_2(n) = NLF_2[u_2(n-1), p_2]$$
$$= \begin{cases} \left\lceil 2^N \times \dfrac{u_2(n-1)}{p_2} \right\rceil & if \;\; 0 \le u_2(n-1) \le p_2 \\ \left\lfloor 2^N \times \dfrac{2^N - u_2(n-1)}{2^N - p_2} \right\rfloor + 1 & if \;\; p_2 < u_2(n-1) < 2^N \end{cases} \qquad (4)$$

The discrete control parameter p_1 is ranging from 1 to $2^{N-1} - 1$ for the PWLCM map, and p_2 is a discrete control parameter ranging from 1 to 2^N -1 for the Skew tent map.

V. Experimental results of the proposed cryptosystem.

A. Secret Key space

To resist the brute force attack, the encryption scheme should have a large secret key space. The size of the secret key of the proposed chaotic generator is formed by four initial conditions (2 for the PWLCM and Skew tent maps and 2 for the *LFSR*), and two parameters. The precision N is equal to

32 bits. The precision of the first and second *LFSR* is L_1=21, and L_2=23. So, the key space is: $3N + N - 1 + L_1 + L_2 = 171$ bits. Therefore, the key space of the secret key is large enough to resist to all kinds of brute force attacks.

B. Computation Times

An experiment was carried out for all encryption algorithms on a unique MATLAB platform and PC with processor Intel 1.83 GHz and RAM memory of 1GB. For example, the computation time required for encrypting 'Lena' image with size 128 x 128 x 3 and for *r* = 6 iterations is respectively 73.728 seconds using AES algorithm in mode CBC, 42.786 seconds using ECKBA and 1.765 seconds using our algorithm. Then, after tested many images with different sizes, the proposed algorithm is at least 41 times faster than AES algorithm, and 24 times faster than ECKBA algorithm.

C. Histogram analysis

The encrypted image should possess certain random properties in order to resist the statistical attacks which are quite common nowadays. The most important property is that the histogram of the encrypted image should be uniformly distributed. The plain and ciphered images of Lena, and their respective histograms are shown in figure 3 and figure 4. As we can see in figure 4.b) the histogram of the ciphered image Lena is uniform.

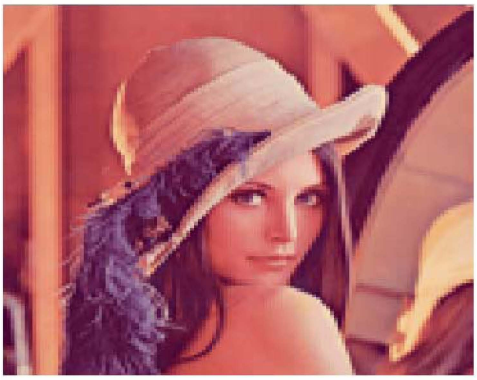

Fig. 3. a) Plain image Lena

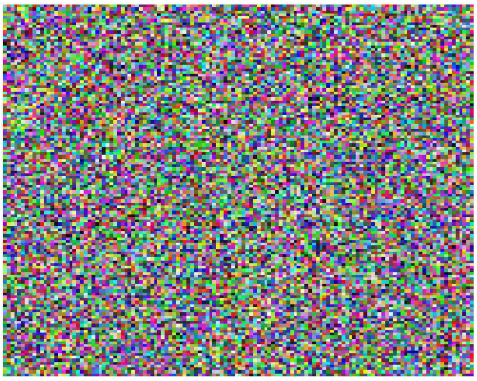

Fig. 3. b) Ciphered image Lena

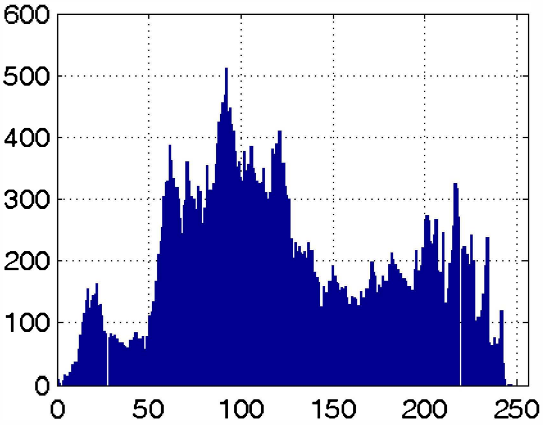

Fig. 4.a) Histogram of Plain image Lena

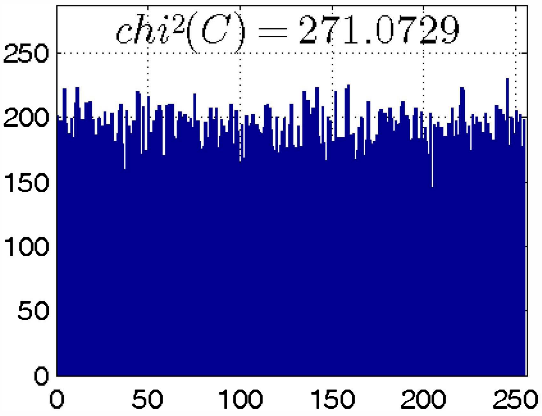

Fig. 4.b) Histogram of ciphered image Lena

D. Correlation analysis

As a general requirement for all the image encryption schemes, the encrypted image should be greatly different from its original form. The correlation analysis is one of usual ways to measure this property. Indeed, it is well known that adjacent pixels in plaintext images are very redundant and correlated. So, in the encrypted images, adjacent pixels should have a redundancy and a correlation as low as possible.

To test the correlation between two adjacent pixels, the following procedure was carried out. Firstly, a thousand pairs of two adjacent pixels are selected randomly in vertical, horizontal, and diagonal directions from the original and encrypted images. And then, the correlation coefficient was computed according to the following formulas:

$$\rho_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

where

$$\operatorname{cov}(x,y) = E\{\ [x - E(x)]\ [y - E(y)]\}$$

$$\textit{and}\ E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i\ ,\ D(x) = \frac{1}{N}\sum_{i=1}^{N}[x_i - E(x)]^2$$

where x , y are gray-level values (vectors) of two adjacent pixels in the image. The obtained results indicate that the correlation coefficient, in all directions, of plain image are close to one, and the correlation coefficient of the encrypted image are close to zero. This means that no detectable correlation exists between the original and its corresponding cipher-image. In figure 5 a) and b) we give respectively the correlation of horizontal adjacent pixels of the plain and ciphered images.

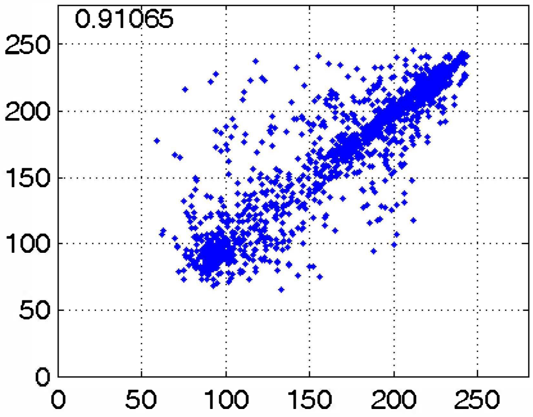

Fig. 5.a) Correlation of adjacent pixels of plain Lena

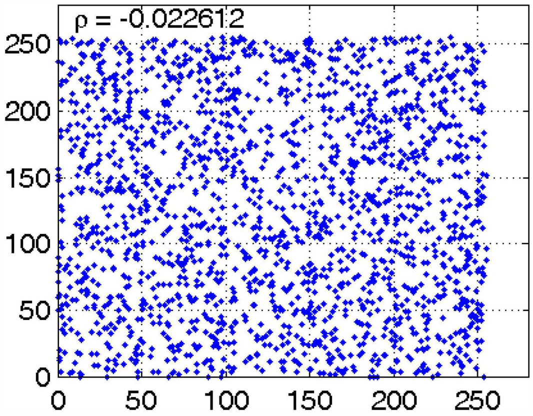

Fig. 5.b) Correlation of adjacent pixels of ciphered Lena

VI. Conclusion

Most of data transmission applications demand both real-time performance and security. In such computer systems, using only AES or ECKBA encryption algorithms may lead to overload due to their prohibitive computational complexity. So, for consumer applications, we need to use a less time consuming encryption algorithm as one proposed in this paper.

The proposed cryptosystem can resist differential, statistical and brute-force attacks. The experimental and theoretical analyses prove that this cryptosystem offers acceptable security levels and very low computational complexity, compared with the AES, and ECKBA algorithms.

This permits the computing system to become underloaded and consequently realize all required secure communications.

By applying the Deadline Mechanism described in section III and using a specific scheduling strategy, we can optimize the Quality of Service by choosing the appropriate encryption algorithm in accordance to the current load and deadline constraints.

References

[1] C.L. Liu and J.W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment,” J. ACM, vol. 20, n°. 1, pp. 46-61, Jan. 1973.

[2] A.L. Liestman and R.H. Campbell, A Fault-Tolerant Scheduling Problem,” IEEE Trans. Software Eng., vol. 12, no. 11, pp. 1089-1095, Nov. 1986.

[3] H. Chetto and M. Chetto, “Some Results of the Earliest Deadline Scheduling Algorithm,” IEEE Trans. Software Eng., vol. 15, no. 10, pp. 1261-1269, Oct. 1989.

[4] S. El Assad (85%), H. Noura (15%), “French Patent : FR2958057 Generator of Discrete Chaotic Sequences with almost Infinite Orbits”, Mars 2010, Extension PCT.

[5] H. Noura, S. El Assad, C. Vladeanu, D. Caragata, “An Efficient and Secure SPN Cryptosystem Based on Chaotic Control Parameters”, IEEE, 6th International Conference for Internet Technology and Secured Transactions, ICITST-2011, Abu Dhabi, December 2011, pp. 226-231.

[6] S. Li, X. Mou and Y. Cai, “Improving security of a chaotic encryption approach”, Phys Lett A 290 (2001), pp. 127–133

[7] G. Chen, Y. Mao, C. Chui, “A symetric image encryption scheme based on 3d chaotic cat maps”, Chaos, Solitons & Fractals, 2004, pp. 749-761.

[8] H.S. Kwok and W.K.S. Tang, “A fast image encryption system based on chaotic maps with finite precision representation”, Chaos, Solitons & Fractals 32 (2007), pp. 1518–1529.

[9] S. Lian, J. Sun and Z. Wang, Security analysis of a chaos-based image encryption algorithm, Phys Lett A 351 (2005), pp. 645–661.

[10] D. Socek, S. Li, S. S. Magliveras, B. Furht, “Enhanced 1-D Chaotic Key Based Algorithm for Image Encryption,”IEEE, Security and Privacy for Emerging Areas in Communications Networks, 2005.

[11] A kumar, Mk Ghose. “Extended substitution-diffusion based image cipher using chaotic standard map”, Commun Nonlinear Sci Numer Simulat(2010),doi:10.1016/j.cnsns.2010.04.010.

[12] B. Schneier, 1996, “Applied Cryptography — Protocols, Algorithms, and Source Code”, C. John Wiley & Sons, Inc., New York 2nd edition..

[13] C.E Shanon, 1949 , Bell System Technical Journal. 28 “Communication theory of secrecy systems”), pp. 656–715.