A lightweight chaos-based cryptosystem for dynamic security management in real-time overloaded applications

Maryline Chetto*

IRCCyN Institute,
University of Nantes,
1 Rue de la Noé, 44321 Nantes, France
E-mail: maryline.chetto@irccyn.ec-nantes.fr
*Corresponding author

Safwan El Assad and Mousa Farajallah

IETR Institute,
University of Nantes,
La Chantrerie, 44306 Nantes cedex 03, France
E-mail: safwan.elassad@univ-nantes.fr
E-mail: mousa.farajallah@univ-nantes.fr

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 to 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 first contribution is a chaos-based cryptosystem which provides an acceptable level of security while executing at least 41 times faster than the AES algorithm. Secondly, we propose a scheduling scheme known as the deadline mechanism in order to guarantee an acceptable level of security even in computing overload situations. Dynamically choosing the best possible cryptosystem in dependance with processing availability is the basis of our mechanism.

Keywords: security; real-time constraints; scheduling; periodic tasks; wireless sensor networks; WSN; chaos-based cryptosystem.

Reference to this paper should be made as follows: Chetto, M., El Assad, S. and Farajallah, M. (2014) 'A lightweight chaos-based cryptosystem for dynamic security management in real-time overloaded applications', *Int. J. Internet Technology and Secured Transactions*, Vol. 5, No. 3, pp.262–274.

Biographical notes: Maryline Chetto received the degree of Docteur de 3$^{\text{ième}}$ Cycle in Control Engineering and the degree of Habilitée à Diriger des Recherches in Computer Science from the University of Nantes, France, in 1984 and 1993, respectively. She is currently a Professor with the Institute of Technology of the University of Nantes. She is conducting her research at IRCCyN Institute. She has published more than 100 journal articles and conference papers in the area of real-time operating systems. Her current research interests include scheduling and power management for real-time energy harvesting applications.

Safwan El Assad joined the University of Nantes, France, in September 1987. He is currently an Associate Professor at Ecole Polytechnique de l'Université de Nantes. His current research area is chaos-based information hiding and security. That includes chaos-based crypto and cryptocompression systems for secure transmitted and stored data, and chaos-based watermarking and stegangraphy. He has graduated nine PhDs and 20 master students. He worked on four European projects and he has published (as an author or co-author) three patents, 24 international journals, four book chapters and 92 articles in international conferences.

Mousa Farajallah has been a PhD student at the University of Nantes since October 2012. His research topic relates to chaos-based crypto-compression systems and hash functions for secure transfer of images and videos.

This paper is a revised and expanded version of a paper entitled 'How to guarantee *secured transactions* with QoS and real-time constraints' presented at 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012), London, UK, 10–12 December 2012.

1 Introduction

There exists a growing number of systems that have real-time and security requirements because sensitive data and processing require special safeguard and protection against unauthorised access. In particular, a variety of motivating real-time applications that run on wireless sensor networks (WSN) need security protections to completely fulfil their security-critical requirements. Each node may have several optional encryption algorithms where each one is assigned a corresponding security level. A high level of security can generally be obtained using the strongest yet slowest encryption function among the alternatives. Computation overhead, i.e., processing utilisation created by encryption mainly depends on complexity of the underlying cryptographic algorithms and size of data to be protected.

An increasing number of applications demand both security and real-time performance. In a real-time system, the computation results must be delivered within time bounds referred to as deadlines. A real-time application is normally composed of multiple programmes called tasks. Hard real-time tasks cannot miss any deadline because it may result in fatal errors. In contrast, firm real-time tasks can miss some deadlines and the system can still work correctly. Examples of hard real-time applications include aircraft control, radar for tracking missiles, and medical electronics. Online transaction processing systems are examples of firm real-time applications. The central issue in the design of any real-time computing system concerns scheduling. For a given set of tasks, the main goal of a scheduling algorithm is to determine an execution order according to which the requirements of each task are satisfied. In this paper, we will investigate the problem of scheduling independent hard real-time tasks that call for encryption functions and execute on a uniprocessor architecture. Sometimes, it may be impossible to execute all the tasks under their timing constraints. One solution to handle the overload problem in firm real-time systems is to reject some of the tasks in order to generate a feasible schedule for the rest. As every task must be completed before deadline in a hard real-time system, one way to avoid missing deadlines is to trade the quality of computation results for timeliness with the use of redundant programmes. The idea of the so-called deadline mechanism is that each task executes one among two encryption algorithms respectively named primary and alternate.

We will describe how to implement the deadline mechanism so as to optimise the global quality of security. We recommend our chaos-based cryptosystem for the alternate since it provides an acceptable level of security with a very short execution time while the AES algorithm can be chosen as the primary algorithm since it supplies the computing system with the highest quality of security with a longer execution time. We will propose a scheduling scheme based on dynamic priority to guarantee either the primary or alternate encryption algorithm of each critical task to be completed in time. Moreover, such a scheme has to complete as many primaries as possible. The strategy statically preallocates time intervals to the alternates. At runtime, primaries are executed first while alternates are executed only if necessary because of processor overload.

The rest of the paper is organised as follows: Section 2 presents typical algorithms for implementing encryption with different QoS requirements. In Section 3, we present the experimental comparative results obtained by three distinct encryption algorithms. Section 4 briefly reviews real-time systems and the associated scheduling algorithms. In Section 5, we 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. The paper concludes with Section 6.

2 Encryption

2.1 General description of the proposed encryption scheme

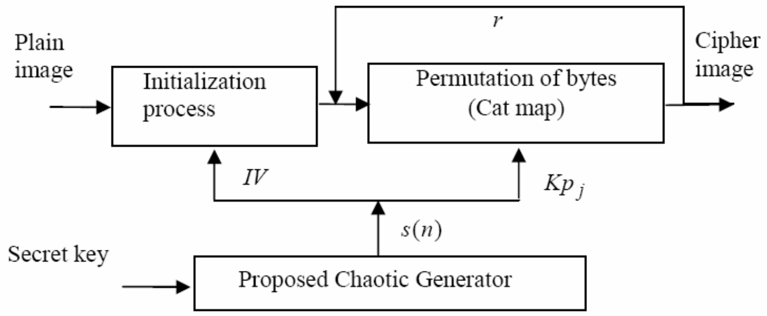
In this section, we suggest three encryption/decryption algorithms for secure data transaction. Two of them are very well known from the literature. It is a question of the advanced encryption standard (AES) algorithm (Schneier, 1996) and the enhanced 1D chaotic key-based algorithm for image encryption (ECKBA). The third one that we propose here, concerns a very fast and robust scheme for image encryption including a strong integer chaotic generator that produces robust pseudo chaotic integer numbers (El Assad, 2012; Li et al., 2001; Kwok and Tang, 2005). Hence, the proposed crypto-system is built using an initialisation layer followed by a 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 efficient and faster than AES and 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 unauthorised person to break this crypto-system. Furthermore, it is immune to statistical, differential, chosen/known-plaintext attacks and brute force attack. Figure 1 gives the principle of the encryption scheme.

In contrast to 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 the proposed encryption scheme runs as follows.

First, we read the image and express it with a decimal matrix I of size L × C × 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 1D line IL with length L × C × P. Then, we apply one time the initialisation process on IB as described in IV-B. Then, we apply the permutation process for each iteration, using the 2D cat map.

Figure 1 Principle of the proposed encryption scheme



2.2 Initialisation process

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

$$IB = \{B_1, B_2, \cdots, B_k, \cdots, B_{nb}\} \tag{1}$$

with

$$nb = L \times C \times P / 4 \tag{2}$$

Thus, the output of this process is given by

$$\begin{cases} BI_k = B_k \oplus IV \text{ if } \operatorname{mod}(k, 2) \neq 0 \\ BI_k = \operatorname{mod}\left(B_k + IV, 2^N\right) \text{ if } \operatorname{mod}(k, 2) = 0 \end{cases} \tag{3}$$
$$k = 1, 2, \cdots, nb$$

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

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

2.3 Permutation process based on cat map

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

$$\begin{bmatrix} Ml' \\ Mc' \end{bmatrix} = Mod\left(\begin{bmatrix} 1 & u \\ v & 1+u\times v \end{bmatrix}\times\begin{bmatrix} Ml \\ Mc \end{bmatrix}\begin{bmatrix} rl+rc \\ rc \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix}\right) \tag{4}$$

where *Ml* and *Mc* are square matrixes given by the following forms:

$$Ml = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 2 & 2 & & 2 \\ \vdots & \ddots & & \vdots \\ M & M & \cdots & M \end{pmatrix}; Mc = \begin{pmatrix} 1 & 2 & \dots & M \\ 1 & 2 & & M \\ \vdots & \ddots & & \vdots \\ 1 & 2 & \cdots & M \end{pmatrix} \tag{5}$$

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

The structure of the dynamic permutation key is

$$Kp = [Kp_1 \,\|\, Kp_2 \,\|\cdots\|\, Kp_r] \tag{6}$$

with

$$Kp_j = [u_j, v_j, rl_j, rc_j] \quad j = 1,\cdots,r \tag{7}$$

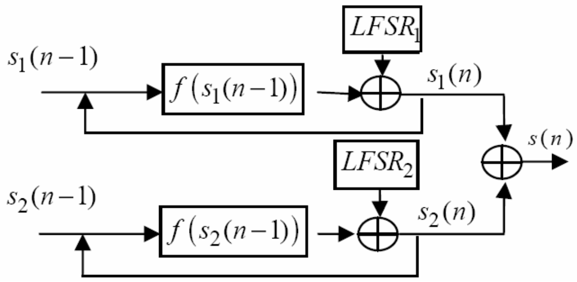
Parameters *u*, *v*, *rl*, *rc* are selected as follows: $0 \le u, v, rl, rc \le M - 1 = 2^q - 1$, where $q = \log_2(M)$ is the necessary number of bits to represent each of the four parameters. If *q* is odd, then after each iteration, the last bytes with length equal to $(L \times C \times P - M^2)$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

2.4 Proposed chaotic generator

The proposed chaotic generator consists in two perturbed discretised chaotic maps (PWLCM and skew tent) connected in parallel as seen in Figure 2 (El Assad and Noura, 2010). The two generated chaotic sequences s_1 i = 1, 2 and s_2 are then bitwise exclusive-or to give the dynamic IV vector and the dynamic keystream Kp_j, j =1, …, r.

Figure 2 Proposed chaotic generator



We apply a perturbation technique in order to extend the periodicity and to control it, as in previous work (Noura and El Assad, 2011). The disturbance is applied using the last k bits of the linear feedback shift registers (*LFSR*) outputs.

The discrete equations of the PWLCM and the skew tent maps are given below:

$$s_1(n) = NLF_1\left[u_1(n-1), p_1\right]$$
$$= \begin{cases} \left\lfloor 2^N \times \dfrac{u_1(n-1)}{p_1} \right\rfloor & \text{if } 0 \le u_1(n-1) < p_1 \\ \left\lfloor 2^N \times \dfrac{2^N - u_1(n-1)}{2^N - p_1} \right\rfloor & \text{if } p_1 \le u_1(n-1) < 2^{N-1} \\ NLF_1\left[2^N - u_1(n-1)\right] & \text{otherwise} \end{cases} \tag{8}$$

$$s_2(n) = NLF_2\left[u_2(n-1), p_2\right]$$
$$= \begin{cases} \left\lceil 2^N \times \dfrac{u_2(n-1)}{p_2} \right\rceil & \text{if } 0 \le u_2(n-1) \le p_2 \\ \left\lfloor 2^N \times \dfrac{2^N - u_2(n-1)}{2^N - p_2} \right\rfloor + 1 & \text{if } p_2 < u_2(n-1) < 2^N \end{cases} \tag{9}$$

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.

3 Experimental results

3.1 Secret key space

The encryption scheme should have a large secret key space so as to resist the brute force attack. The size of the secret key of the proposed chaotic generator is formed by four initial conditions (two for the PWLCM and Skew tent maps and two for the *LFSR*) and two parameters. Precision N is equal to 32 bits. The precision of the first and second LFSR is $L_1 = 21$ and $L_2 = 23$. Thus, 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.

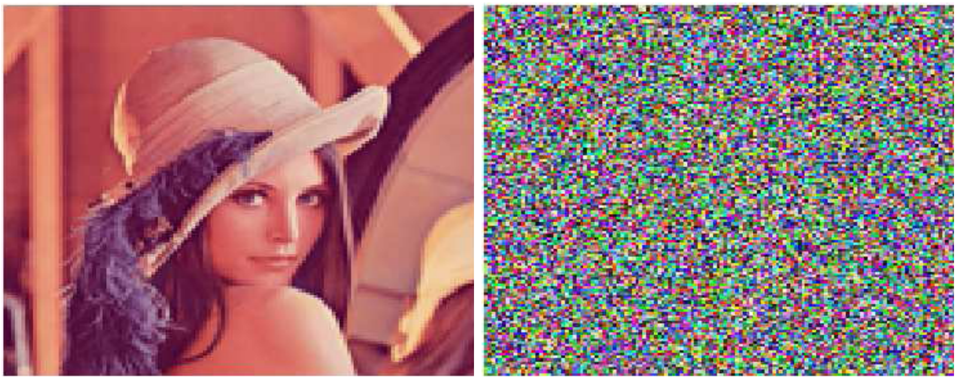
3.2 Computation times

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 1 GB. For example, the computation time required for encrypting 'Lena' image with size 128 × 128 × 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. After testing many images with different sizes, the proposed algorithm proves to be at least 41 times faster than AES algorithm and 24 times faster than ECKBA algorithm.

3.3 Histogram analysis

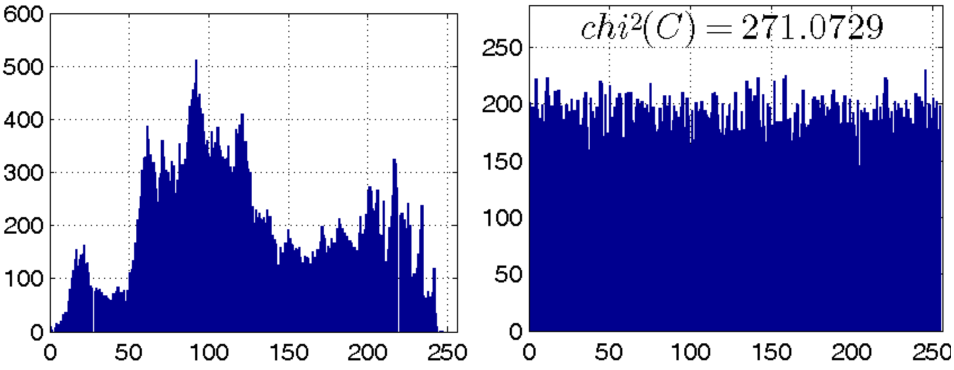
The encrypted image should possess certain random properties in order to resist the statistical attacks which are quite common nowadays. The most important property is that the histogram of the encrypted image should be uniformly distributed. The plain and ciphered images of Lena and their respective 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.

Figure 3 Plain and ciphered images, (a) plain image Lena (b) ciphered image Lena (see online version for colours)



(a) (b)

Figure 4 Histograms of plain and ciphered images, (a) histogram of plain image Lena (b) histogram of ciphered image Lena (see online version for colours)



(a) (b)

3.4 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 the usual ways to measure this property. Indeed, it is well-known that adjacent pixels in plaintext images are very redundant and correlated. So, adjacent pixels in the encrypted images should have as low as possible redundancy and correlation. The following procedure was carried out to test the correlation between two adjacent pixels. 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 is computed according to the following formulae (Song et al., 2012):

$$\rho_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)\sqrt{D(y)}}} \tag{10}$$

where

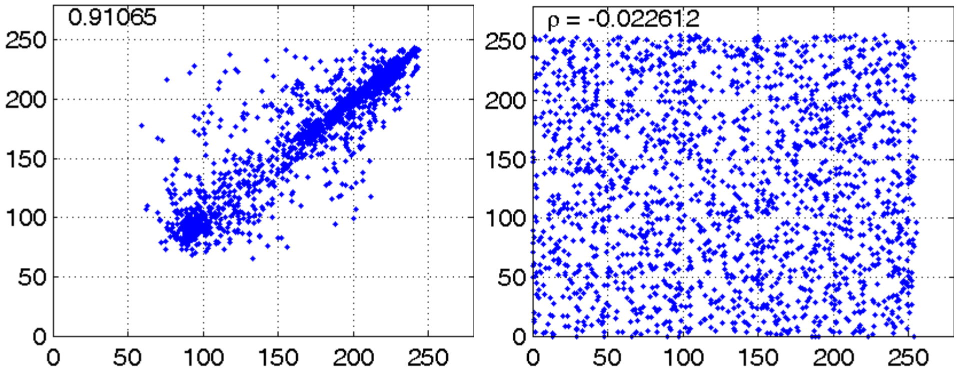
$$\operatorname{cov}(x, y) = E\{[x - E(x)][y - E(y)]\} \tag{11}$$

and

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

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

Figure 5 Correlation of adjacent pixels, (a) correlation of adjacent pixels of plain Lena (b) correlation of adjacent pixels of ciphered Lena (see online version for colours)



(a) (b)

4 Real-time computing and security

“The correctness of real-time applications depends not only on the logical computation being performed but also on the time at which the results are produced”. A real-time system is a computing system that is designed to handle workloads whose tasks have completion deadlines. In contrast to conventional computer systems where the goal usually is to minimise task response times, the emphasis is on satisfying the timing constraints of tasks. Recent years have seen the emergence of WSN that must support high data rate and real-time sensing of physical environments. For example, a very active area of research concerns WSNs that serve to monitor the health of patients. Typically, a node of the network reports data to a base station which analyses data and alerts health care personnel upon alarm detection.

Security is nowadays of critical importance for a wide range of real-time applications including WSNs. These modern real-time networks require high security level to give confidentiality of information stored in packages delivered through wireless links. In order to achieve the goal of meeting all task deadlines, the designers of safety critical real-time systems typically attempt to anticipate every eventuality and incorporate it into the design of the system. Such a system would, under ideal circumstances, never miss deadlines and its behaviour would be as expected by the system designers. In reality, however, overload conditions may occur wherein the required processing load exceeds the system capacity, thereby resulting in missed deadlines. If this happens, it is important that the performance of the system degrades gracefully.

When the real-time system is in charge of computing or/and transmitting secure data, we need for a specific scheduler which reveals capable of achieving high quality of security for data and making the best effort to guarantee real-time requirements. There are a number of ways both to detect and to handle overloads. A detection method should be predictive. A predictive technique would build a schedule and then determine if a task will miss its deadline if executed under that schedule. Another issue concerns what actions to take when an overload is detected. Possibilities include aborting tasks which are thought to provoke the overload and/or executing alternative tasks. The following section examines the issue of executing alternative tasks in charge of encryption.

5 Security and scheduling issues

There are many challenges for real-time system security, such as challenges related to efficient implementations and schedulability. The basic idea of our security model is that real-time tasks can flexibly select encryption algorithms to form an integrated security scheme. Static schedulability analysis can be used to predict whether the timing constraints of a real-time system can be met in the worst case situation. One of the most commonly used schedulability tests for real-time systems is the schedulability test for the earliest deadline first (EDF) scheduler where the highestpriority task has the closest deadline. An EDF scheduler dispatches tasks based on their relative deadline. The EDF policy is optimal for independent preemptable tasks on single processor meaning: If a set of independent preemptable tasks are schedulable onto a single processor by any scheduler, then they are also schedulable by an EDF scheduler (Liu and Layland, 1973). To verify whether a set of tasks are schedulable under EDF, we can use a very simple and robust schedulability test. The test is to check whether the utilisation ratio of the task set in the underlying real-time system is less than 1.

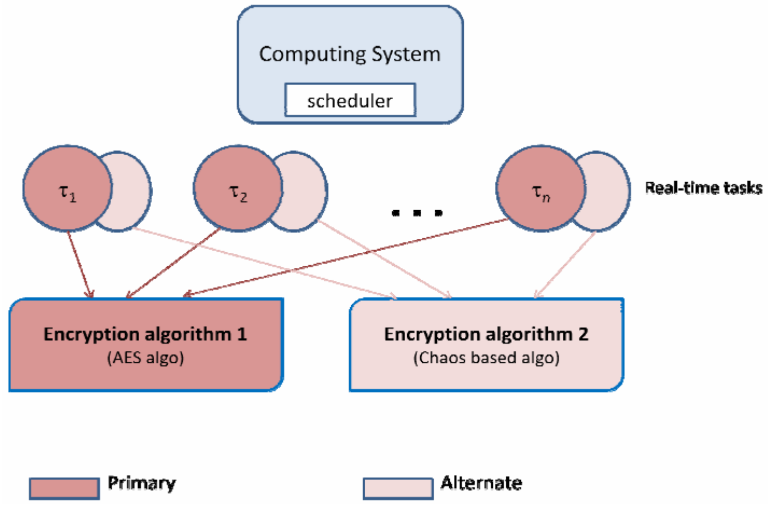
Computation overhead caused by encryption mainly depends on the cryptographic algorithm used and size of data to be protected. In accordance to the cryptographic algorithms' performance, each algorithm is assigned a security level. In case the timing constraints cannot be met (i.e., a timing fault cannot be avoided), one way of meeting the deadlines is to trade computation quality (measured in terms of security level) for timeliness.

6 The deadline mechanism

6.1 Description

The so-called deadline mechanism (Liu and Layland, 1973) combines two methods to provide software timing fault-tolerance in hard real-time task systems. Each task has two versions respectively called primary and alternate. The primary version contains more functions and produces good quality results but its exact execution time can be prohibitive. The alternate version contains only the minimum required functions and produces less precise results but its worst case execution time can be estimated precisely. If the primary of a task fails (due to manifestation of a bug for example) during its execution or if its successful completion cannot be guaranteed (due to insufficient processor time), the alternate of the task must be activated.

Figure 6 Security management with the deadline mechanism (see online version for colours)



The challenge in the implementation of the deadline mechanism is two-fold:

1 how to guarantee that either the primary or the alternate version of each task be completed in time

2 and how to successfully complete as many primaries as possible.

6.2 Scheduling framework

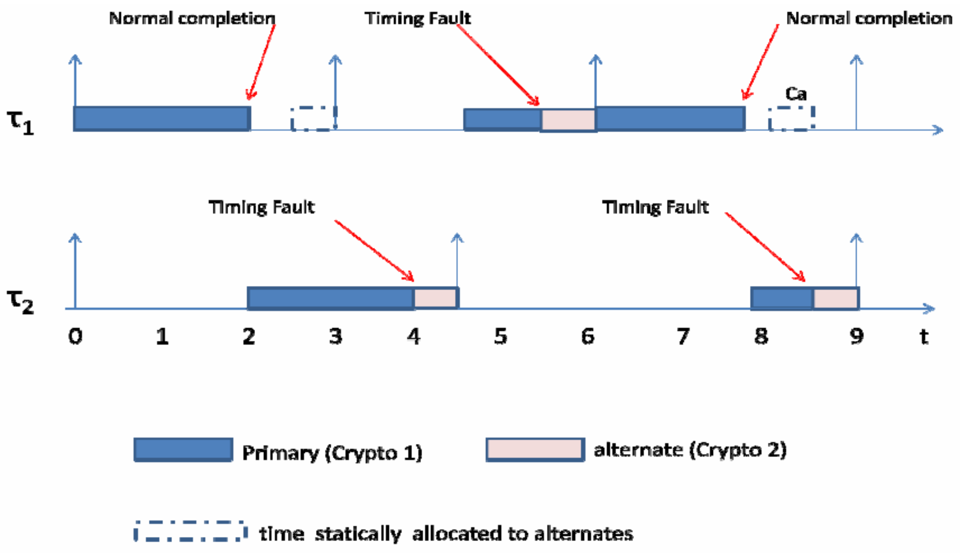
Chetto and Chetto (1989) proposed the last chance strategy to maximise the number of successful primaries. An offline strategy reserves time intervals for the alternates. Each interval is chosen so that any alternate starts execution at the latest possible time instant.

At runtime, primaries are scheduled during the remaining intervals before their respective alternates. Any alternate may preempt a primary when a time interval reserved for the alternates is reached. Whenever a primary completes successfully, the execution of its corresponding alternate is no longer needed. Hence, an online mechanism has to dynamically deallocate the time interval(s) reserved for the alternate so as to increase the processor time available for the execution of other primaries. In other words, maximum processor time is made available as soon as possible for the primaries. And this leads to maximise their chance of success.

6.3 Illustrative example

We consider a periodic task set consisting of two tasks τ_1 and τ_2. Each task τ_i is described by a period T_i between two instances of a task. Deadlines coincide with periods. We have T_1 =3 and $T_2 = 4.5$. An invocation of a task is called a job and the k^{th} invocation of task τ_i is denoted $\tau_{i,k}$. Here, the first jobs of tasks τ_1 and τ_2 are released simultaneously. Each task needs data encryption. We assume that each task has a primary version that calls for the best but time-consuming encryption algorithm (such as AES or ECKBA). The exact execution time of the primary is not known a priori because it depends notably on the size of the data to encrypt. In the other hand, each task has an alternate version with a very short and well known worst case execution time, $\mathrm{Ca_i}$. Here, $\mathrm{Ca_1} = 0.5$ and $\mathrm{Ca_2} = 0.5$. We note that the task set is feasible since the processor utilisation, given by $\mathrm{Ca_1/T_1} + \mathrm{Ca_2/T_2}$ is approximately equal to 28%.

Figure 7 Illustration of the deadline mechanism (see online version for colours)



At time 0, primary of $\tau_{1,1}$ has the closest deadline. It executes successfully and completes at time 2. Thus, the time reservation between 2.5 and 3 for the alternate of $\tau_{1,1}$ is removed. $\tau_{2,1}$ starts at time 2 and is discarded at time 4.5 which corresponds to the latest start time of the alternate of $\tau_{2,1}$ which executes and completes at time 4.5. Primary $\tau_{1,2}$ then executes and is discarded because alternate $\tau_{1,2}$ has to start and completes at time 6. Primary $\tau_{1,3}$ executes successfully. Primary $\tau_{2,2}$ fails at 8.5 and a secure result is obtained by its alternate at time 9.

In that example, five jobs have been released. Two of them were capable to make use of the best encryption function with the highest security level. Three of them were obliged to call for the back-up encryption function with a lower but nevertheless acceptable security level.

7 Conclusions and future works

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 a less time consuming encryption algorithm as one proposed in this paper. The 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 realise all required secure communications.

This paper has presented a scheduling framework for uniprocessor real-time systems that require data encryption. As the AES and ECKBA encryption algorithms involve long execution times for the real-time tasks, we propose to manage a possible resulting overload by online commuting on a degraded encryption mode. Our proposition amounts to call for another less time-consuming encryption algorithm such as our chaos based algorithm, whenever necessary. The advantage of such a dynamic mechanism is both to prevent from deadline violations and to produce results with an acceptable level of security at any time. The so called deadline mechanism based on a form of software redundancy was initially dedicated, in the 80s, to fault-tolerance in control applications. Such a mechanism has proved its utility for security optimisation of real-time data with adaptive and selective cryptosystems.

In future works, we will address the question of security in new generation 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.

The introduction of energy harvesting capabilities in networks has introduced additional design questions. How to intelligently use the ambient incoming energy to optimise the quality of service 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? As in the present paper, we are proposing the deadline mechanism to cope with processing overload and in addition energy shortage.

Acknowledgements

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

References

Chetto, H. and Chetto, M. (1989) 'Some results of the earliest deadline scheduling algorithm', *IEEE Transactions on Software Engineering*, October, Vol. 15, No. 10, pp.1261–1269.

El Assad, S. (2012) 'Chaos BSED information hiding and security', in *International Conference for Internet Technology and Secured Transactions*, IEEE, London, December, pp.67–72.

El Assad, S. and Noura, H. (2010) 'Generator of chaotic sequences and corresponding generating system', WO/2011/121218, 6 October.

Kwok, H. and Tang, W.K.S. (2005) 'A fast image encryption system based on chaotic maps with finite precision representation', *Chaos, Solitons and Fractals*, November, Vol. 32, No. 4, pp.1518–1529.

Li, S., Mou, X. and Chui, C. (2001) 'Improving security of a chaotic encryption approach', *Physics Letters A*, November, Vol. 290, No. 3, pp.127–133.

Liu, C-L. and Layland, J.W. (1973) 'Scheduling algorithms for multiprogramming in a hard real-time environment', *Journal of the Association for Computing Machinery*, Vol. 20, No. 1, pp.46–61.

Noura, H. and El Assad, S. (2011) 'An efficient and secure SPN cryptosystem based on chaotic control parameters', in *2011 International Conference for Internet Technology and Secured Transactions (ICITST)*, Abu Dhabi, 11–14 December, pp.226–231.

Schneier, B. (1996) *Applied Cryptography*, 2nd ed., John Wiley & Sons, New York, NY, USA.

Song, C-Y., Qiao, Y-L. and Zhang, X-Z. (2012) 'An image encryption scheme based on new spatiotemporal chaos', *Optik*, October.