Non-Linear behavior of Encryption Algorithm under windows Environment

Naji M. Soheeb

Diala University

Ammar J. Fatah

Science Gate company, software development company.

Abstract :

This paper is devoted to present a new parameter that should be included along the design phase of and modular EA that will implemented on windows operating system. This parameter is the Tick Resolution (Tic) which represents how frequent the system will leave current process and give the control to other, this will be used to compromised EA and dig into integral black box. A mathematical model has been presented to formulate the relationship between Tic and the key parameters of EA which the complexity, this paper will prove the nonlinearity of the behavior of EA due to high tick resolution, where increasing the complexity might lead to worse security , an optimal solution has been proposed with other solutions. A practical test has been included to present the effectiveness of Tic on the EA linearity.

1-introduction

Windows operating systems have been the market leader over more than two decades, and according to Microsoft, they already sold more than billion of original copies of various versions of windows operating systems (i.e. windows NT, windows XP, windows 2000, windows 2003 server and other versions). Microsoft introduced windows as a multitask and interactive operating system, where, windows can run more than one process at a certain time and give the user the feeling of being running concurrently. windows was designed from the beginning to server all classes of users ranging from novice users to professional users, windows can be used by kids and old people as it could be used by scientist or network administrator. Microsoft has equipped windows with wide spectrum of applications that fulfill users' needs in a very wide range of fields (i.e office work, multimedia, network … etc), thus, windows is in every where.

Windows security was and still gaining a great attention of windows users due to critical and valuable information stored and managed by windows. The horizon of the threats has been expanded and fall into Two main sectors:

1. **System security**
2. **Data security.**

System security is the back door of the data security, for an example key log which collect user keyboard clicks, this will ruin any data security implemented within that system even if it uses the most powerful security tools.

Data encryption is a very well known tools used to secure data stored and exchanged through the computer network. Data encryption by itself expanded to a wide spectrum of kinds varying according to the task it was used for. Encryption algorithms are the core of Data encryption and it's the real measure of how secure the data will be.

Encryption algorithms which are based on computation complexity are graded to how much time and space it takes to do the job. It well known that *more processing time will impose more security,* this paper will show that under windows environment, increasing computation complexity will weakening the overall security of the algorithm, in other words the algorithm will behave non-linearly.

2- The architecture of computer operating system

The operating system could be defined as the peaceful environment for a process or processes to be executed.

In that manner operating system can be seen as :

First: Extended Machine where the operating system presents the machine (computer) to the user (process) as an equivalent machine, this isolates the user(process) from what is going on under the beneath. As an example is how operating system write data on the sectors on a hard disk , a user (process) does not care about , it is the operating system responsibility. Extended machine is not only an abstracted level of the machine , it is sometimes the Virtual machine or logical machine for a process or a user. [1,4]

Second: Resource Manager where the operating system is providing the user (process) with an orderly and controlled allocation of the system resources. System resources is a wide range of system components and its account and function are varies corresponding to the system it lives within. As example of resources : processors, timers, disks, mice, network interfaces, printers, … etc. [1,2,7]

3- Concept of process and threads

All the runnable software on the computer, sometimes including the operating system, is organized into a number of *sequential processes*, or just processes. A process is just an executing program, including the current values of the program counter, registers, and variables. [1,4]

In multiprogramming each process has its own virtual CPU. In reality the real CPU switches back and forth from process to process. [1]

With the CPU switching back and forth among the processes, the rate at which a process performs its computation will not be uniform and probably not even reproducible if the same processes are run again. *Thus , processes must not be programmed with built-in assumptions about timing.[1,p72]*

To keep track of each of the different processes that are executing concurrently in memory , the operating system creates and maintains a block

of data for each process in the system. This data block is known as a *process control block*, frequently abbreviated as PCB. [2,5]

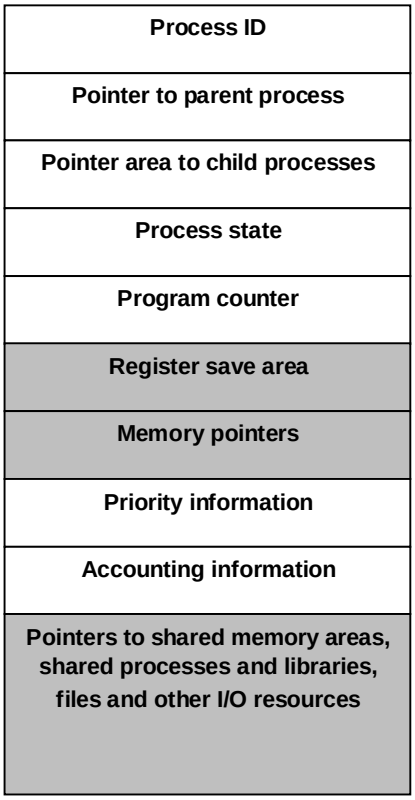

Figure 1 , PCB structure

The important fields that this paper is focusing on are the shaded ones which it will used later to access the data stored in memory before processing it by the encryption algorithm. When context switching occurs , CPU will store its registers in the memory as it is been shown in figure 1, registers will be used later to access data and code.

4- Threads

A thread represents a piece of a process that can be executed independently of other parts of the process. Each thread has its own context, consisting of a program counter value, register set , and stack space, but shares program code, and data , and other system resources .

Context switching among threads is easier for the operating system to manage because there is no need to manage memory, files, and other resources and no need for synchronization or communication within the process.[2,4,6]

This paper will focus on the process itself rather than the thread due to the fact that the thread is piece of the process.

5- CPU scheduling

CPU scheduling provides mechanisms for the acceptance of processes into the system and for the actual allocation of CPU time to execute those processes. A fundamental objective of multitasking is to optimize use of the computer system resources, both CPU and I/O by allowing multiple processes to execute concurrently. [2,4]

A key issue related to scheduling is when to make scheduling decisions. It turns out that there are a variety of situations in which scheduling is needed.

First: when a new process is created

Second: when a process exits

Third: when a process blocks on I/O.

Fourth: when an I/O interrupt occurs.[2, 4,5]

This paper is focusing on the fourth type of forcing the CPU to make a decision to Schedule. A key kernel of the scheduler is the *Dispatcher*. A dispatcher select .

6- Tick resolution

tick resolution is how frequent the operating system update its timestamp, applications that depend on time change could be triggered right a way. The timestamps that you can obtain from Windows NT are limited to a maximum resolution of 10 or 15 milliseconds, depending on the underlying hardware. This means that the minimum time slice (quantum) the could be allocated to a process is 10 or 15 millisecond and along that time there is no possible normal way to take the control of that process from monopolizing the CPU time,

in other words, at that 10 or 15 millisecond the process can't be interrupted in a normal way. By using performance counters in conjunction with the system time to calculate smaller time increments to be able to contact a thread or

perform some other task at intervals more frequent than 10 milliseconds [3],(implementing high resolution timer). This paper has adopted the technique which has been used in reference [3] which enhances time resolution to reach 1 mSec.

7- Encryption algorithm and CPU time slice

Encryption is the Art of transforming messages(plain data) into less intelligible to all not intended receiver. Essentially, the encryption of data employs such methods in various communication devices to establish communication security, authentication and digital signatures.

Encryption is a special form of computation and such systems depend upon the difficulty of computation for their security. This paper is focusing of one type which the public key encryption where there are two keys one private and the other is public.

This major type of encryption system (i.e RSA cipher) enciphers a message M by:

$$C = M \bmod N \quad \text{----------} \quad (1)$$

$$M = C \bmod N \quad \text{----------} \quad (2)$$

Where:

C: is the cipher

M: is the Message

e: encryption (private) key

d: decryption (public) key

p: large prime number

q: large price number relative to p

N: (p-1)(q-1)

Execution time =nInst x CPI x Clock Time ---------- (3)

nInst: Number of Instruction

CPI: clockticks/instructions, where each instruction has a certain number of ticks.

Clock time (Tc) = 1/ Clock rate. ---------- (4)

Thus,

EAE is a function of Tc ,

EAE = f(Tc) = nInst x Tc ---------- (5)

Where, EAE is encryption algorithm execution

Real Time = Execution Time + Latency ---------- (6)

8- Mathematical model

(Encryption Algorithm Integrity under multitasking Environment)

To develop a scheme that presents the relation ship between EA's parameters (time, space) and windows imposed standards (Tick resolution), some definitions have to be adopted :

1. **EA has to behave as constant in term of time and space , in other words , EA has the same complexity at any given stage..**
2. **Space is to be ignored due to the fact that modern systems could be equipped with a very large memory , some times , it reaches Tera bytes**
3. **EA is at least a process implemented on the system and it is represented as a series of code**

$$EA = \sum_{I=1}^{T.P.C} Ci \qquad \text{----------} \qquad (7)$$

Where:

Ci: is the ith code of the program.

T.P.C: Total Program Code

4- EA is compromised when there is an ability to divide the execution of the EA , in other words,

$$EA = \sum_{I=1}^{K} Ci + \sum_{K+1}^{T.P.C} Ci \qquad \text{----------} \qquad (8)$$

Where k is a stop and resume point within T.P.C

5- CPU is available to execute a code only over Tic, Tic is Tick Resolution and it is a propriety of an operating system , i.e. windows NT 10 millisecond., Unix and linux 100 nSec.

6- EA linearity is represented by the following equation

$$EAL = \frac{nInst \times Tc}{1 - \sin(\theta)} \qquad \text{----------} \qquad (9)$$

$$\theta = (int(Tp / Tic) \times \pi / 2) \qquad \text{----------} \qquad (10)$$

Thus,

$$Tp = n.Tic + Tcs. \qquad \text{----------} \qquad (11)$$

Where Tcs is the time for context switching , since the CPU will idle of processing along Tcs, we can omit it,

$$Tp = n\ Tic \rightarrow Tp/Tic = n. \qquad \text{----------} \qquad (12)$$

n is number of taps that we can insert it into the encryption algorithm implementation.

9- Experimental Results

The following results collected by using RSA optimized code and run it with different Tick resolutions.

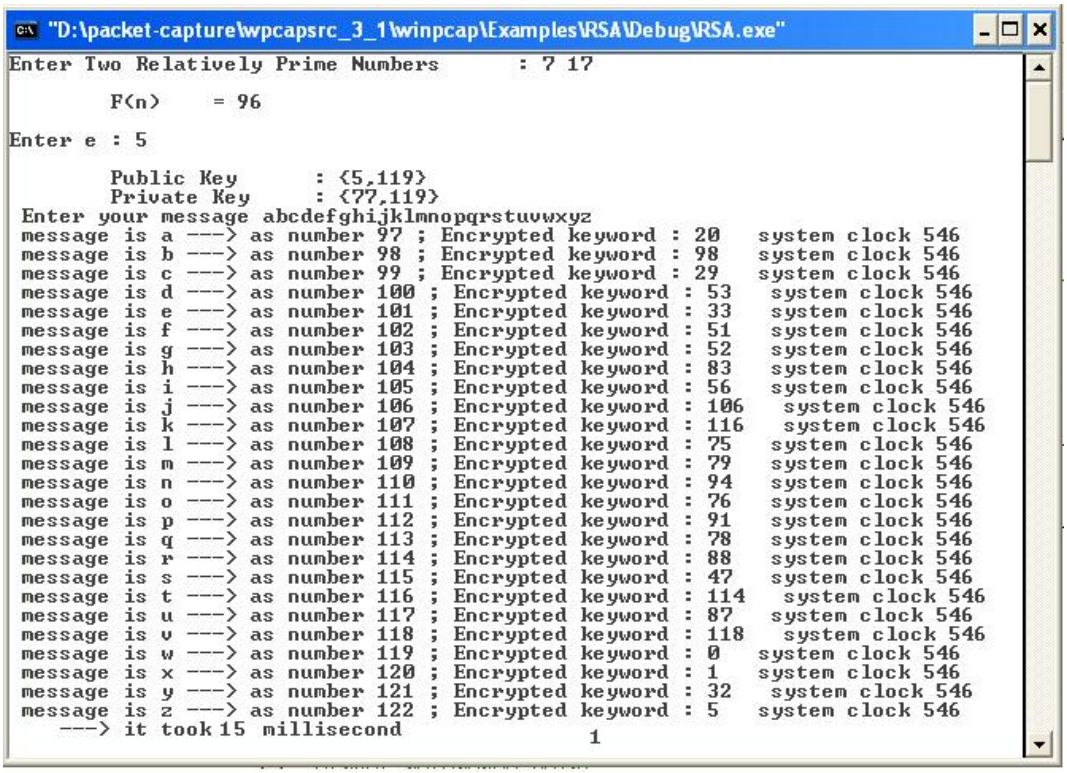

Figure (2), RSA at 10 millisecond resolution

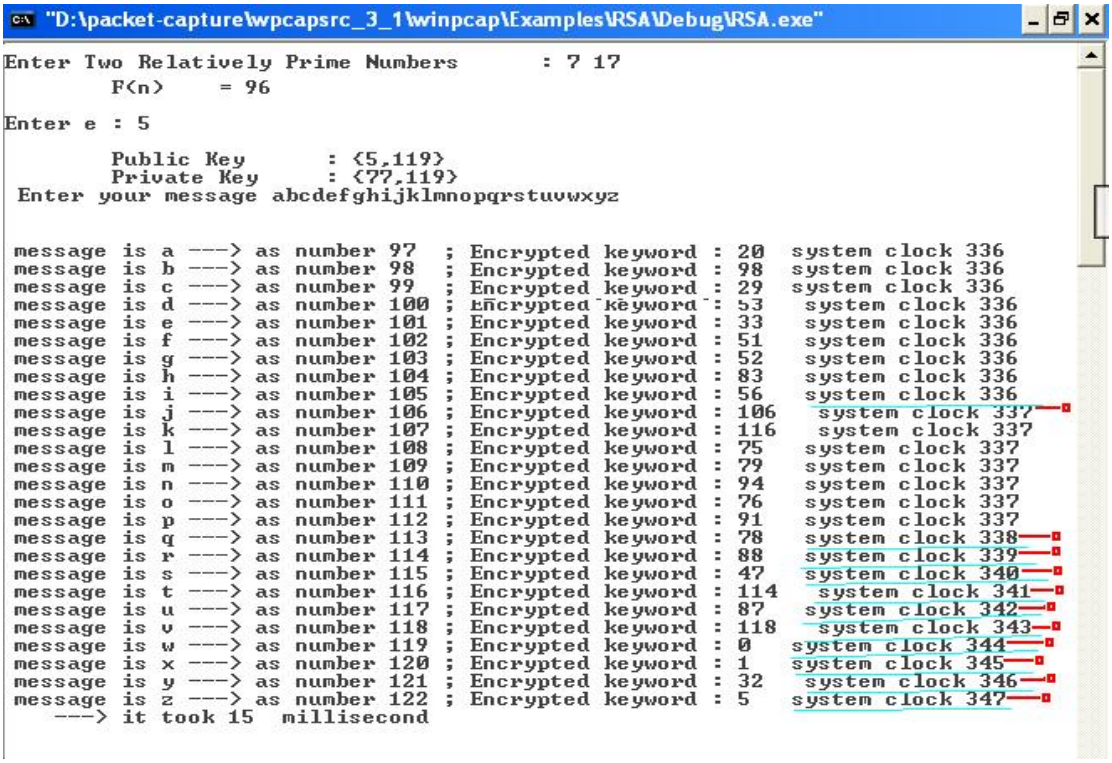

Figure (3), RSA at I millisecond resolution, process has been interrupted

14 times

Figure (2) and 3) are examples snapshot to show what is going on when the complexity of the EA has been increased , in other words higher Tick resolution.(eq.12).

In figure (3) each marked line is data been read from the memory before it has been sent to the EA, where a context switching was occurring and the EA was stopped and the control been taken by another process.

The data structure which was defined in Figure(1) , PCB, has been used to identify the targeted data for certain Process ID

10- Proposed solutions

In multitasking environment (i.e. windows XP/2k/2003),Its impossible to fully own the CPU from a certain process and prevent other processes from getting their slices, this would turn the multitasking environment into single task and for sure stop the beauty of Interactive GUI(because GUI is running as a separate process to interact human generated events), such as clicking a mouse or typing on the keyboard.

This paper is devoted to present the Tick Resolution effect of EA , and as a secondary task it proposes the following solution's performance were measured according to equ. 9,10 :

Optimal solution: Using embedded hardware, in this solution a full feature microcomputer card will be embedded into the system. This microcomputer card will have its own processor and all the resourced it needs to complete a process such as EA. The results, in this case the encryption data, will be send to the main host, the computer , through the standard bus (i.e. USB , PCMCIA or any other high speed bus ports).

From the main processor point of view :

$$Tic = Tp \quad \text{----------} \quad (13)$$

Feasible solution: this paper is using a method of changing the hardware RTC, you can minimize the impact of accessing the real-time clock (RTC) by replacing the hardware RTC with a software RTC, eliminating calls to the RTC, or accessing the clock at optimal times.

In a standard x86 bus architecture, where the CMOS houses the RTC and the CMOS resides on the bus, the RTC accesses the bus several times. In doing so, the bus can be locked for several microseconds, blocking it from other activities.

This translates into higher latencies for ISRs and longer times to fully service a process (i.e. EA process). The EA software should be designed from the first place to use this technique or it could be added later, the modern approaches in software engineering grant this ability (code reuse, COM programming ,... etc)

Administrative solution: Windows Administrator can give a hand and force low Tick Resolution on the windows operating system, that will give the EA the chance to finish its core process without interruption. This solution will count on the security kernels within windows to prevent other non Authorized personal or process to change it back to its normal resolution.

11- What is new in this paper

Till this writing, all Encryption Algorithms are valid for all platforms and the only issue is the hardware requirement, so, it dose not matter if you have windows xp , windows NT or other type of platform since you have efficient machine.

This paper presented a mathematical model that connected platform characteristic (i.e. each operating system has its own policy to manage RTC) to Encryption Algorithm complexity and draw for the first time a new restriction that it can't be passed over safely without taking the whole system into chaos, as a security point of view.

12- Conclusions

1- **A new parameter has to be considered when designing EA to be implemented under windows environment, which is the Tic (tick resolution).**
   The key is :

   Tr <= Tick.

2- **Tick Resolution can be used as a key attack parameter against EA implemented under windows.**

3- **Windows can't take the tick resolution lower due to the lack of performance it could be imposed by that. This is why there is a limiting number of processes / threads to share CPU execution time.**

13- References

1- Andrew S. Tanenbaum, Modern Operating Systems, Second Edition, 2001, Prentice-Hall, Inc. Upper Saddle River, New Jersey.

2- Ben Smith and Brian Komar with the Microsoft Security Team, 2003, Microsoft Press.

3- Corbalan, J., Martorell, X., and Labrarta, J., "Performance-Driven Processor Allocation", Proc. Fourth Symp. On Operating Systems Design and Implementations, USENIX, pp. 59-71,2000

4- DEITEL, " JAVA How To Program",2005,Prentic-Hall.

5- IRV Englander, "The Architecture of computer hardware and systems software", 2003, John Wiley & Sons, Inc.

6- Jones, O. "Introduction to the X window System, Englewood Cliffs", 2000, Prentice Hall.

7- Walter Oney," windows secrets", 2007,McGraw Hill Inc.

التصرف اللاخطي لخوارزميات التشفير في بيئة النوافذ

عمار عبد الجبار فتاح
شركة بوابة العلم لتطوير البرامجيات

م. ناجي مطر سحيب
جامعة ديالى – قسم الحاسبات

االمستخلص:

ان هذا البحث مكرس لعرض متغير جديد يجب ان يوخذ بنظر الاعتبار عند تصميم خوارزمية التشفيرللعمل في بيئة النوافذ , هذا المتغير هو سرعة دقات مزامن الوقت، حيث انه يمثل عدد المرات التي سوف يترك النظام العملية الحالية التي يقوم بها ويذهب لتنفيذ عملية اخرى. ان هذا المزامن سوف يكون مفتاح مهاجمة امنية خوارزمية التشفير من خلال الدخول الى شفرة عملها والتي يجب ان تكون صندوق اسود لا يمكن الدخول اليه . سوف يتم طرح نموذج رياضي يصف علاقة سرعة المزامن مع وقت تنفيذ خوارزمية التشفير، حيث ان كفاءة الخوارزمية يتم الحكم عليها من خلال تعقيد الخوارزمية والذي هو مرتبط بشكل وثيق بالزمن اللازم لتنفيذ الخوارزمية هذا البحث سوف يبرهن بطلان القاعدة المعروفة ان زيادة تعقيد الخوارزمية يؤدي الى زيادة امنيتها ، هذا البرهان ينطبق على الانظمة التي تعمل في بيئة النوافذ او الانظمة التي تعمل باسلوب تعدد المهام. هذا البحث سوف يطرح حلول مثالية لهذه المشكلة وحلول اخرى مقبولة بعد ان يقوم بعرض النتائج المختبرية التي تؤكد صحة تاثير سرعة المزامن على كفاءة الخوارزمية