

## **Non-Linear behavior of Encryption Algorithm under Windows Environment**

**Naji M. Sahih, Bashar Talib Hameed**

**Diyala University- Collage of Science- Computer .Dept, Diyala University- Collage of  
Science- Math. Dept**

**Receiving Date: 22-1102-2010 - Accept Date: 1714-036-2011**

### **Abstract**

This paper is devoted to present a new parameter that should be included along the design phase of and modular Encryption Algorithm 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 compromise Encryption Algorithm and that must be an integral black box.

A mathematical model has been presented to formula the relationship between Tic and the key parameters of Encryption Algorithm which the complexity, This paper will prove the nonlinearity of the behavior of Encryption Algorithm due to high tick resolution, where increasing the complexity might lead to lack in security. An optimal solution have been proposed practical test to present the effectiveness of Tic on the Encryption Algorithm linearity.

## Non-Linear behavior of Encryption Algorithm under Windows Environment

### 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 serve all classes of users ranging from novice 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 user's needs in a very wide rang of field (i.e. office work, multimedia, network ... etc), thus, windows is in every where [1,4].

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 collects user keyboard clicks, this will ruin and 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

### Non-Linear behavior of Encryption Algorithm under Windows Environment

weakening the overall security of the algorithm, in other words the algorithm will behave non – linearly [1,4].

## 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 present 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, 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 resource. System resource is a wide range of system components and its account and function are varies corresponding to the system it lives within. As example of resource:

Processors, timers, disks. Mice, network interface, printers, etc. [1,2,7]

## 3- concepts 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]

### Non-Linear behavior of Encryption Algorithm under Windows Environment

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.*[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 control block, frequently abbreviated as Process control block? [2,5]

|                                                                                         |
|-----------------------------------------------------------------------------------------|
| Process ID                                                                              |
| Pointers to parent process                                                              |
| Pointer area to child processes                                                         |
| Process state                                                                           |
| Program counter                                                                         |
| Register save area                                                                      |
| Memory pointers                                                                         |
| Priority information                                                                    |
| Accounting information                                                                  |
| Pointers to shared memory areas, shared processes and libraries, and other I/O resource |

Fig(1) PCB structure

The important fields that this paper is focusing on are the shaded ones which it will use 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 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 resource.

### Non-Linear behavior of Encryption Algorithm under Windows Environment

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 threads due to the fact that the threads are piece of the process.

#### 5- CPU scheduling

**CPU** scheduling provides mechanisms for the acceptance of processes into 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 resource, 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) 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 others words, at that 10 or 15 millisecond the process

### Non-Linear behavior of Encryption Algorithm under Windows Environment

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 **1mSec**.

### 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[4]

Encryption is a special form of computation and such systems depend upon the difficulty of computation for their security. This paper is focusing on one type which is 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 :[5]

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

And decipher it by:

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

Where :

$C$  : is the cipher

$M$  : is the message

$e$  : encryption (private) key

$d$  : decryption (public) key

$p$  : large prime number

$q$  : large prime number relative to  $p$

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

# Non-Linear behavior of Encryption Algorithm under Windows Environment

$$\text{Execution time} = \text{no of Inst} \times \text{CPI} \times \text{clock Time} \quad \text{-----} \quad (3)$$

**nInst:** Number of Instruction

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

$$\text{Clock time (Tc)} = 1/\text{Clock rate.} \quad \text{-----} \quad (4)$$

Thus,

EAE (Encryption Algorithm Execution) is a function of Tc,

$$\text{EAE} = f(\text{Tc}) = \text{nINST} \times \text{Tc} \quad \text{-----} \quad (5)$$

Where, EAE is encryption algorithm execution

$$\text{Real Time} = \text{Execution Time} + \text{Latency} \quad \text{-----} \quad (6)$$

## 8- Mathematical model

(Encryption Algorithm Integrity under multitasking Environment)

To develop a scheme that present the relationship between **EA's** parameters (time, space) and widows 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 Tera bytes.
- 3- EA is at least a process implemented on the system and it is represented as a series of code.

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

# Non-Linear behavior of Encryption Algorithm under Windows Environment

Where:

$C_i$  : is the  $i$ th 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,

$$EA = \sum_{i=1}^k C_i + \sum_{k+1}^{T.P.C} C_i \quad \text{-----} \quad (8)$$

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

- 5- CPU is available to execute a code only over  $T_{ic}$ ,  $T_{ic}$  is Tick Resolution and it is 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.

$$EA = \frac{\text{no of Inst} \times T_c}{1 - \sin(\theta)} \quad \text{-----} \quad (9)$$

$$\theta = (\text{int}(T_p / T_{ic}) \times \pi / 2) \quad \text{-----} \quad (10)$$

Thus :

$$T_p = n.T_{ic} + T_{cs} \quad \text{-----} \quad (11)$$

$T_p$ : Time for a process.

Where  $T_{cs}$  is the time for context switching, since the CPU will idle of processing along  $T_{cs}$ , we can omit it,

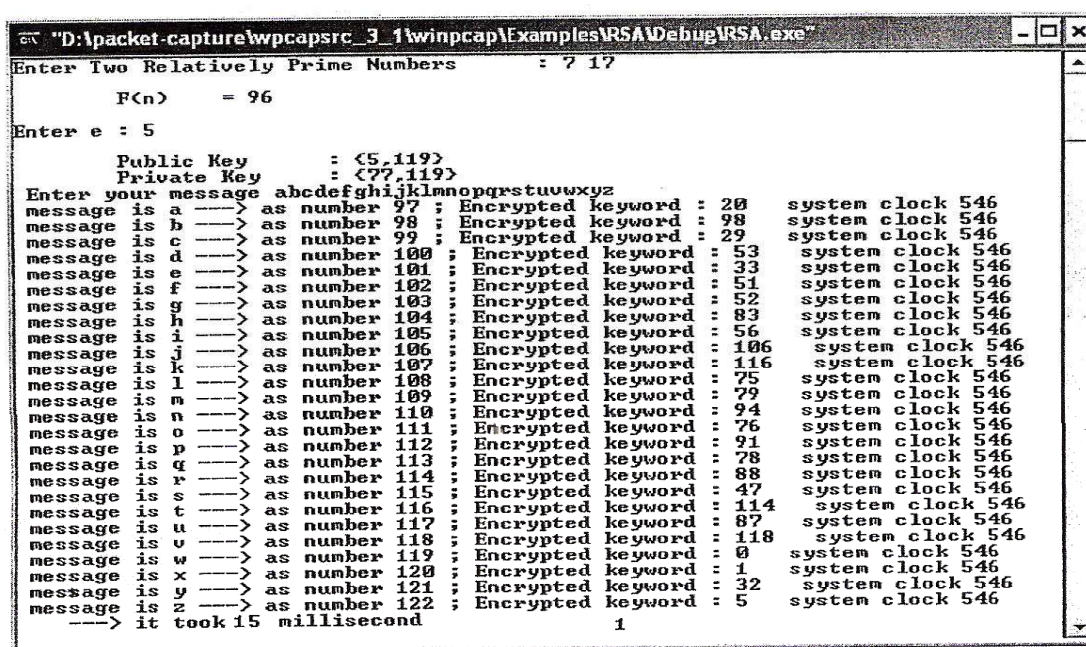
$$T_p = n.T_{ic} \quad \xrightarrow{T_p} \quad T_p / T_{ic} = n. \quad \text{-----} \quad (12)$$

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

## Non-Linear behavior of Encryption Algorithm under Windows Environment

### 9- Experimental Result

The following results collected by using RSA optimized code and run It with different Tick resolution.



```

D:\packet-capture\wpcapsrc_3_1\winpcap\Examples\RSA\Debug\RSA.exe
Enter Two Relatively Prime Numbers : 7 17
      P(n)      = 96
Enter e : 5
      Public Key      : (5,119)
      Private Key     : (77,119)
Enter your message abcdefghijklmnopqrstuvwxyz
message is a ---> as number 97 ; Encrypted keyword : 20      system clock 546
message is b ---> as number 98 ; Encrypted keyword : 98      system clock 546
message is c ---> as number 99 ; Encrypted keyword : 29      system clock 546
message is d ---> as number 100 ; Encrypted keyword : 53      system clock 546
message is e ---> as number 101 ; Encrypted keyword : 33      system clock 546
message is f ---> as number 102 ; Encrypted keyword : 51      system clock 546
message is g ---> as number 103 ; Encrypted keyword : 52      system clock 546
message is h ---> as number 104 ; Encrypted keyword : 83      system clock 546
message is i ---> as number 105 ; Encrypted keyword : 56      system clock 546
message is j ---> as number 106 ; Encrypted keyword : 106     system clock 546
message is k ---> as number 107 ; Encrypted keyword : 116     system clock 546
message is l ---> as number 108 ; Encrypted keyword : 75      system clock 546
message is m ---> as number 109 ; Encrypted keyword : 79      system clock 546
message is n ---> as number 110 ; Encrypted keyword : 94      system clock 546
message is o ---> as number 111 ; Encrypted keyword : 76      system clock 546
message is p ---> as number 112 ; Encrypted keyword : 91      system clock 546
message is q ---> as number 113 ; Encrypted keyword : 78      system clock 546
message is r ---> as number 114 ; Encrypted keyword : 88      system clock 546
message is s ---> as number 115 ; Encrypted keyword : 47      system clock 546
message is t ---> as number 116 ; Encrypted keyword : 114     system clock 546
message is u ---> as number 117 ; Encrypted keyword : 87      system clock 546
message is v ---> as number 118 ; Encrypted keyword : 118     system clock 546
message is w ---> as number 119 ; Encrypted keyword : 0       system clock 546
message is x ---> as number 120 ; Encrypted keyword : 1       system clock 546
message is y ---> as number 121 ; Encrypted keyword : 32      system clock 546
message is z ---> as number 122 ; Encrypted keyword : 5       system clock 546
---> it took 15 millisecond      1
  
```

Figure ( 2), RSA at 10 millisecond resolution

### Non-Linear behavior of Encryption Algorithm under Windows Environment

```

D:\packet-capture\wpcapsrc_3_1\winpcap\Examples\RSA\Debug\RSA.exe
Enter Two Relatively Prime Numbers : 7 17
P(n) = 96
Enter e : 5
Public Key : <5,119>
Private Key : <77,119>
Enter your message abcdefghijklmnopqrstuvwxyz

message is a --> as number 97 ; Encrypted keyword : 20 system clock 336
message is b --> as number 98 ; Encrypted keyword : 98 system clock 336
message is c --> as number 99 ; Encrypted keyword : 29 system clock 336
message is d --> as number 100 ; Encrypted keyword : 53 system clock 336
message is e --> as number 101 ; Encrypted keyword : 33 system clock 336
message is f --> as number 102 ; Encrypted keyword : 51 system clock 336
message is g --> as number 103 ; Encrypted keyword : 52 system clock 336
message is h --> as number 104 ; Encrypted keyword : 83 system clock 336
message is i --> as number 105 ; Encrypted keyword : 56 system clock 336
message is j --> as number 106 ; Encrypted keyword : 106 system clock 337
message is k --> as number 107 ; Encrypted keyword : 116 system clock 337
message is l --> as number 108 ; Encrypted keyword : 75 system clock 337
message is m --> as number 109 ; Encrypted keyword : 79 system clock 337
message is n --> as number 110 ; Encrypted keyword : 94 system clock 337
message is o --> as number 111 ; Encrypted keyword : 76 system clock 337
message is p --> as number 112 ; Encrypted keyword : 91 system clock 337
message is q --> as number 113 ; Encrypted keyword : 78 system clock 338
message is r --> as number 114 ; Encrypted keyword : 88 system clock 339
message is s --> as number 115 ; Encrypted keyword : 47 system clock 340
message is t --> as number 116 ; Encrypted keyword : 114 system clock 341
message is u --> as number 117 ; Encrypted keyword : 87 system clock 342
message is v --> as number 118 ; Encrypted keyword : 118 system clock 343
message is w --> as number 119 ; Encrypted keyword : 0 system clock 344
message is x --> as number 120 ; Encrypted keyword : 1 system clock 345
message is y --> as number 121 ; Encrypted keyword : 32 system clock 346
message is z --> as number 122 ; Encrypted keyword : 5 system clock 347
--> it took 15 millisecond
  
```

Figure (3), RSA at 1 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), process control block(PCB), has been used to identify the targets data for certain Process identifier.

## Non-Linear behavior of Encryption Algorithm under Windows Environment

### 10- Proposed solutions

In multitasking environment (i.e. windows X/ 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 equation 9 and 10.

**Optimal solution** : Using embedded hardware, in this solution a full feature microcomputer card will be embedded in to the system, this microcomputer card will have its own processor and all the resources it needs to complete a process such as EA. The results, In this case the encryption data, will be sent 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:

$$T_{ic} = T_p \text{ ----- (13)}$$

**Feasible solution** : this paper is using a method of 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 ISR, and longer times to fully service a process (i.e. EA process). The EA software should 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).

### Non-Linear behavior of Encryption Algorithm under Windows Environment

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

This paper resented 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 tie 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- Conclusion

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

$$Tr \leq \text{Tick}$$

- 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.

**Non-Linear behavior of Encryption Algorithm under  
Windows Environment**

**13- References**

- 1- Andrew S. Tanenbuam, "Modern operating Systems", Second Edition, 2001, Prentice – Hall, Inc. Upper Saddle River, new Hersey.
- 2- Ben Smith and Brain Komar with the Microsoft Security Team, 2003, Microsoft press.
- 3- Corbalan, J., Martorell, X., and Labrarta, J., "Performance – Driven Processor Allocation ", Proc. Fourth Sump. 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.