Research Journal of Applied Sciences 10 (2): 66-69, 2015
ISSN: 1815-932X

Evaluation of Time Complexity in Investigation of Methods of Modular Exponentiation

[1]Aigul K. Shaikhanova, [2]Aleksandr D. Zolotov, [2]Eldos M. Muhametov,
[3]Mikolaj P. Karpinski and [2]Dinara O. Kozhahmetova
[1]Kazakh National Technical University Named after K.I. Satpayev,
22a Satpayev Street, 050013 Almaty, Republic of Kazakhstan
[2]Semey State University Named after Shakarim,
20a Glinka Street, 071410 Semey, Republic of Kazakhstan
[3]Academy of Technologies and the Humanities in Bielsko-Biala,
2 Willowa Street, 43-309 Bielsko-Biala, Poland

Abstract: Basic concepts of evaluation of algorithm complexity such as time and space complexity were considered by its consumption of basic computer resources. Comparative investigation of operations of binary, β and sliding window method of modular exponentiation with "left-to-right" and "right-to-left" reading bits of exponent was conducted.

Key words: Time complexity, space complexity, modular exponentiation, binary method, β method, sliding window method

INTRODUCTION

Traditionally, it is accepted to estimate the degree of complexity of algorithm in terms of consumed basic computer resources such as CPU time and RAM. In this regard such concepts as time and space complexity of the algorithm are introduced (Bellezza, 2001).

Time required to perform basic operations of exponentiation algorithms: Time complexity parameter is particularly important for applications with interactive mode of program or for real-time control tasks (Biham and Shamir, 1997). It is necessary to spend some time to perform the operations of modular exponentiation algorithms (binary, β, sliding window methods).

Execution time of single operation of the algorithm depends on the speed of processor, so it can be said that in general, every single step of the algorithm is performed during certain time. Basic operations of modular exponentiation algorithms and time spent on each of them can be represented as Table 1.

In general, it can be assumed that ratio between the values of these times is as follows:

$$c \le b \le q \le t \le r \le s \le d \tag{1}$$

Table 1: Time required to perform basic operations of exponentiation algorithms

Operation	Time in ticks	Meaning of the operation
$a = b$	C	Simple assignment
$z = x \bmod n$	B	Modulus assignment
$\text{FIND}(\max\{n_i...n_j\} \mid i-j+1 \le w,\ n_j = 1)$	Q	Finding the longest sequence of bits, so that $i-j+1 \le w$ and $n_j = 1$
$n = (n_{k-1}...n_0)_2$	T	Representation of numbers in binary notation
$y = x \times x \bmod m$	R	Modulus squared
$z = x \times y \bmod m$	S	Modular multiplication
$z = y^{\beta} \bmod m$	D	Modular exponentiation

On the basis of data in Table 1, we can construct a mathematical model for calculating the time required to perform each of the algorithms for the implementation of methods of modular exponentiation. Following time is required to perform binary method; "left-to-right" reading:

$$T1(n) = t + c + \sum_{i=k-1}^{0} r_i + \sum_{\{i|n_i=1\}} s_i = t + c + \lceil \log n \rceil \times r + H(n) \times s \tag{2}$$

"right-to-left" reading:

$$T2(n) = t + c + b + \sum_{\{i|n_i=1\}} s_i + \sum_{i=0}^{k-1} r_i = t + c + b + H(n) \times s + \lceil \log n \rceil \times r \tag{3}$$

Corresponding Author: Aigul K. Shaikhanova, Kazakh National Technical University Named after K.I. Satpayev, 22a Satpayev Street, 050013 Almaty, Republic of Kazakhstan

Following time is required to perform β method; "left-to-right" reading:

$$T3(n,w) = t + c + \sum_{i=1}^{\beta-1} s_i + c + \sum_{i=k-1}^{0} (d_i + s_i) = t + 2c + \left(\frac{\lceil \log n \rceil}{w} + 2^w - 1\right) \times s + \frac{\lceil \log n \rceil}{w} \times d \tag{4}$$

"right-to-left" reading:

$$T4(n,w) = t + b + \sum_{w=1}^{\beta-1} c_w + \sum_{0}^{k-1}\left(d_{\{i|n_i=0\}} + s_{\{i|n_i=1\}} + d_{\{i|n_i=1\}}\right) + 2c + \sum_{w=\beta-1}^{1} 2s_w = t + \left(2^w + 1\right)c + b + \frac{\lceil \log n \rceil}{w} \times d + \left(\frac{\lceil \log n \rceil}{w} - w_0(n) + 2^{w+1} - 2\right) \times s \tag{5}$$

where $w_0(n)$ number of zero bits in the representation of number n to the base β. Obviously that in binary image numbers n∊$\lceil \log n \rceil$-H(n) of zero bits.

To convert number into β notation, binary image is divided into n windows with length of w. Therefore, the upper bound $w_0(n)$:

$$w_0^{max}(n) = \left\lfloor \frac{\lceil \log n \rceil - H(n)}{w} \right\rfloor \tag{6}$$

On the other hand, the lower bound can easily be determined as:

$$w_0^{min}(n) = \left\lfloor \frac{\left(\lceil \log n \rceil - H(n)\right) \times w}{(w-1) \times \lceil \log n \rceil} \right\rfloor \tag{7}$$

Following time is required to perform sliding window method; "left-to-right" reading:

$$\begin{aligned} T5(n,|w_i|) &= b + s + \sum_{j=1}^{2^{|w_i|}-1} s_j + t + 2c + \sum_{i=0}^{k-1}\left((r+c)_{\{i|n_i=0\}} + (q+s+c+r)_{\{i|n_i\neq 0\}}\right) = b + s + \left(2^{|w_i|} - 1\right)s + \\ & t + 2c + \left(k - H(n)\right)(r+c) + p(q+s+c) + r\left(|w_o| + \ldots + |w_i|\right) = t + b + 2c + kr + 2^{|w_i|}s + \\ & p(q+s+c) + \left(k - H(n)\right)c = t + b + \left(2 + p + \lceil \log n \rceil - H(n)\right)c + \lceil \log n \rceil r + \left(2^{|w_i|} + p\right)s + pq \end{aligned} \tag{8}$$

"right-to-left" reading:

$$\begin{aligned} T6(n,|w_i|) &= t + b + \sum_{\{j=1,3,\ldots,2^{|w_i|}-1\}} c_j + c + \sum_{i=k-1}^{0}\left((r+c)_{\{i|n_i=0\}} + (q+s+c+d)_{\{i|n_i\neq 0\}}\right) + \sum_{\{v=2^{|w_i|}-1,\ldots,5,3\}} (2s_v) + c = t + \\ & b + \left(2^{2|w_i|-2} + 1\right)c + \left(k - H(n)\right)(r+c) + p(q+s+c+d) + 2^{2|w_i|-1}s + c = t + b + \\ & \left(2^{2|w_i|-2} + 2 + \lceil \log n \rceil - H(n) + p\right)c + \left(\lceil \log n \rceil - H(n)\right)r + \left(2^{2|w_i|-1} + p\right)s + pq + pd \end{aligned} \tag{9}$$

Where:

p = The number of windows

$(|w_0|+\ldots+|w_i|)$ = The sum of all odd windows that equal to Hamming weight, since these windows consist of only single bits

Obviously that:

$$p_{max} = \left\lceil \frac{\log n}{2} \right\rceil$$

and

$$p_{min} = \left\lceil \frac{H(n)}{w_i} \right\rceil \tag{10}$$

Thus in general for the investigation of the execution time of this algorithm following average value can be used:

$$p = \frac{\left\lceil \frac{H(n)}{w_i} \right\rceil + \left\lceil \frac{\log n}{2} \right\rceil}{2} \tag{11}$$

DETERMINATION OF THE MOST PRODUCTIVE ALGORITHM OF MODULAR EXPONENTIATION

It is obvious that to improve performance of asymmetric encryption devices it is necessary to determine the most productive of known algorithms of modular exponentiation that are used in such devices. We consider solution of this problemon the example of described above binary, β and sliding window methods.

As mentioned above, total execution time of the algorithm of binary method is dependent only on the length of binary image of number n. Execution time of the algorithm of β method depends not only on the length of the binary image of number n but also on the value of β (i.e. on the number w). Execution time of the algorithm of sliding window method depends on the length of the binary image of number n and on the width of odd window. Taking this into account, it is possible to investigate dependence of the execution time of the algorithm on the length of binary image of number n.

Figure 1 shows this dependence for the averaged values of the Hamming weight (H(n)) and number of zeros in the β image of number n ($w_0(n)$) as well as for different values of w, the width of odd window and values of c = 1, b = 1.5, q = 1.6, t = 1.6, r = 15, s = 16, d = 19 (the ratio between the variables correspond to the number of ticks that the processor spends to perform the corresponding operations (Comer and Stevens, 2000).

In this case,T1(n) and T2(n), T3(n, 2) and T3(n, 4) execution time of "left-to-right" β algorithm of modular exponentiation at w = 2 and w = 4, correspondingly. The T4(n, 2) and T4(n, 4) execution time of "right-to-left" β algorithm of modular exponentiation at w = 2 and w = 4, correspondingly. T5(n, 3) and T6(n, 3) execution time of "left-to-right" and "right-to-left" method of sliding window at the length of the window w_i = 3.

Analysis of Fig. 1 shows that the execution time of algorithms of modular exponentiation is linear. In addition, the fastest algorithms are that of "left-to-right" and "right-to-left" β method and the most time consuming is the algorithm of binary method.

Figure 2 and 3 shows, respectively the dependence of speed of algorithms of "left-to-right" and "right-to-left" β method on the value of power of w base at different key length and at averaged value of the Hamming weight.

Using data from Fig. 2 and 3, we can determine the optimum base at which there is the smallest delay in work of algorithm, i.e., the minimum T3 and T4, respectively and thus, provide maximum productivity for given values of the exponent. For algorithms of β method values of w presented in Table 2 will be the best.

Table 2: The optimum values of the power of bases of β method at different length of n key

	w (β method)	
Length of n key	"Left-to-right"	"Right-to-left"
4096	8	7
2048	7	6
1024	6	5
512	6	5
256	5	4

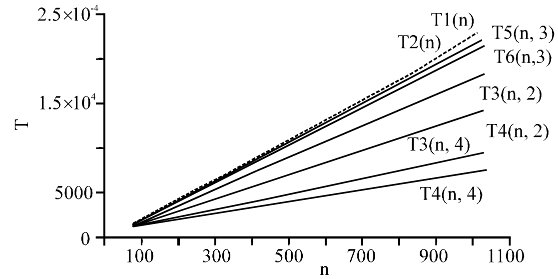

Fig. 1: Evaluation of the performance characteristics of investigated algorithms

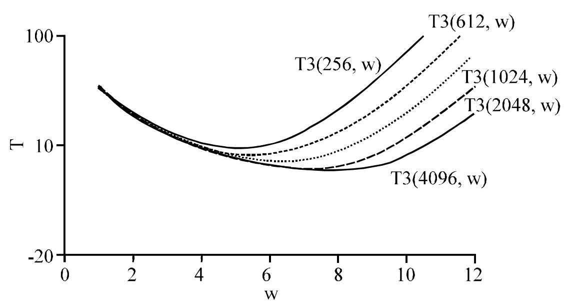

Fig. 2: Dependence of the speed of algorithm of "left-to-right" β method on the value of the power of w base

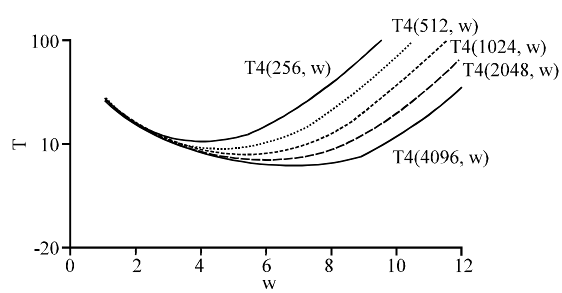

Fig. 3: Dependence of the speed of algorithm of "right-to-left" β method on the value of the power of w base

Space complexity of the algorithm, i.e., consumption of computer memory for its execution, becomes critical when the volume of data to be processed

Table 3: The maximum number of memory cells involved in the execution of algorithms of modular exponentiation

Modular exponentiation algorithm	The number of memory cells
Binary	2
β	2^{w}
Sliding window	2^{w_1}

is almost equal to amount of RAM. In modern computers, acuteness of this problem is reduced due to increase in amount of Random Access Memory (RAM) and to usage of multilevel storage system. For programs that implement the algorithm very large, almost unlimited, memory space (virtual memory) is available. Lack of main memory only leads to a slowdown through the exchange of data with the disk. Special techniques are used to minimize the loss of time in this exchange. It is the usage of cache memory and hardware preview of program commands on the required number of steps ahead that allows to transfer required values from disk to main memory in advance (Kshetri and Murugesan, 2013; Kurose and Ross, 2011). When performing considered modular exponentiation algorithms maximum number of registers according to Table 3 are busy in computer memory.

Analysis of Table 3 shows that the largest consumption of memory is during the execution of sliding window algorithm, since length of the largest window can be equal to the half length of the key. In the case of β modular exponentiation algorithm consumption of memory depends on the chosen notation, i.e., on the value (Mangard *et al*., 2007).

CONCLUSION

Parameters of time and space complexity were investigated. It was found that best methods for application are β method and sliding window method of modular exponentiation with “left-to-right” reading bits of exponent.

REFERENCES

Bellezza, A., 2001. Countermeasures against side-channel attacks for elliptic curve cryptosystems. Cryptology ePrint Archive: Repoet 2001/103, USA., November 2001. https://eprint.iacr.org/2001/103.pdf.

Biham, E. and A. Shamir, 1997. Differential fault analysis of secret key cryptosystems. Proceedings of the17th Annual International Cryptology Conference on Advances in Cryptology, August 17-21, 1997, Santa Barbara, CA., pp: 513-525-.

Comer, D.E. and D.L. Stevens, 2000. Internetworking with TCP/IP, vol III: Client-Server Programming And Applications. Prentice Hall, Upper Saddle River, NJ., ISBN-13: 978-0130320711, Pages: 601.

Kshetri, N. and S. Murugesan, 2013. EU and US cybersecurity strategies and their impact on businesses and consumers. Computer, 46: 84-88.

Kurose, J.F. and K.W. Ross, 2011. Computer Networking: A Top-Down Approach. 6th Edn., Addison-Wesley, Boston, MA.

Mangard, S. and E. Oswald and T. Popp, 2007. Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer, Berlin, Germany.