Int'l J. of Communications, Network and System Sciences, **2011, 4, 357-363**
doi:10.4236/ijcns.2011.46041 Published Online June 2011 (http://www.SciRP.org/journal/ijcns)

Scientific Research

Space Complexity of Algorithm for Modular Multiplicative Inverse

Boris S. Verkhovsky
Computer Science Department, New Jersey Institute of Technology, University Heights, Newark, USA
E-mail: *verb73@gmail.com*
Received April 23, 2011; *revised May* 28, 2011; *accepted June* 14, 2011

Abstract

In certain computational systems the amount of space required to execute an algorithm is even more restrictive than the corresponding time necessary for solution of a problem. In this paper an algorithm for modular multiplicative inverse is introduced and its computational space complexity is analyzed. A tight upper bound for bit storage required for execution of the algorithm is provided. It is demonstrated that for range of numbers used in public-key encryption systems, the size of bit storage does not exceed a 2K-bit threshold in the worst-case. This feature of the Enhanced-Euclid algorithm allows designing special-purpose hardware for its implementation as a subroutine in communication-secure wireless devices.

Keywords: Modular Multiplicative Inverse, Public-Key Encryption, Space Complexity, Tight Upper Bound, Extended Euclid Algorithm, Prefix Coding, Enhanced Euclid Algorithm, Custom-Built Circuits

1. Algorithm for Modulo Multiplicative Inverse

The operation of modular multiplicative inverse is essential for public-key encryption, modular arithmetic [1] and for applications based on the Chinese Remainder Theorem [2].

1.1. Introduction

Operation of multiplicative inverse modulo n is a basic operation in modular arithmetic. A number x is called a *modular multiplicative inverse*, (MMI, for short) of p_1 modulo p_0, [2], if it satisfies equation

$$p_1 x \bmod p_0 = 1. \tag{1.1}$$

1.2. Enhanced-Euclid Algorithm (EEA)

Consider integer variables L, M, S, t and Boolean variable c;

Assign $L := p_0$; $M := p_1$; $c := 0$;
Repeat $t := \lfloor L/M \rfloor$;
$S := L - Mt$; $c := 1 - c$; $L := M$; $M := S$; (1.2)
push t {onto the top of the stack}; (1.3)
until either $S = 1$ **or** $S = 0$;
if $S = 0$, **then** $\gcd(p_0, p_1) = M$; **output** "*MMI does not exist*"; $M = \gcd(p_0, p_1)\}$;
else assign $S := 0$; $M := 1$;
repeat pop t {from the top of the stack};

$$L := Mt + S; \quad S := M; \quad M := L; \tag{1.4}$$

until the stack is *empty*; **output**
if $c = 0$ **then** $MMI := L$ **else** $MMI := p_0 - L$;

for more details see [3,4].

The algorithm is valid for both cases: for $p_0 > p$ or $p_0 < p_1$. In the latter case, assign $p_1 := p_1 \bmod p_0$.

Validity of the EEA is discussed in [3] and its time complexity is analyzed in [4]. Although both analysis and computer experiments demonstrate that the EEA is faster than the Extended-Euclid algorithm [2], the EEA requires the *storage of stack*, {see (1.3)-(1.4)}. The worst-case *space* complexity of the EEA is analyzed in this paper if

$$p_0 > p_1. \tag{1.5}$$

2. Bit-Storage Required for Stack

2.1. Direct Problem

Let $N(p_0, p_1)$ be the number of bits required to store the stack. Find a maximal p_0 such that for *all* values of p_1 satisfying (1.5) the EEA does not require more than s bits for storage of the stack. Consider optimization

problems:

$$Q(s, p_0) := \max_{2 \le p_1 < p_0} N(p_0, p_1) \le s, \quad (2.1)$$

and let

$$q(s) := \max_{p_0} Q(s, p_0). \quad (2.2)$$

2.2. Dual Problem

In order to analyze space complexity of the EEA let's consider a sequence $\{n_0, n_1, \cdots, n_k, \cdots\}$ generated in accordance with the following rules: let $a > b \ge 1$;

$n_{k+1} := q_k n_k + n_{k-1}$, where all $q_k \ge 1$ and $n_1 := a$; $n_0 := b$; and for all $k \ge 1$, n_k are integers. Then for every $k \ge 1$

$$(n_{k+1}, n_k) = (n_k, n_{k-1}) \begin{pmatrix} q_k & 1 \\ 1 & 0 \end{pmatrix} \quad [5]. \quad (2.3)$$

Therefore, for every integer $r \ge 1$

$$(n_{r+1}, n_r) = (a, b) \prod_{k=1}^{r} \begin{pmatrix} q_k & 1 \\ 1 & 0 \end{pmatrix}; \quad (2.4)$$

and $$n_{r+1} = (a, b) \prod_{k=1}^{r} \begin{pmatrix} q_k & 1 \\ 1 & 0 \end{pmatrix} (1, 0)^T.$$

Suppose that a memory that stores an array of quotients $\{q_1, q_2, ..., q_{r-1}, q_r\}$ has restricted capacity. For instance, suppose that it cannot store more than s bits. Consider the following optimization problem:

Find

$$n(r, s) := \min_{\{q_k : k=1,..,r\}} (a, b) \prod_{k=1}^{r} \begin{pmatrix} q_k & 1 \\ 1 & 0 \end{pmatrix} (1, 0)^T \quad (2.5)$$

under constraint

$$\sum_{k=1}^{r} \lfloor \log_2 2q_k \rfloor = s, \quad (2.6)$$

where integers a, b, r and s in (2.5) and (2.6) are specified parameters.

Here $\lfloor \log_2 2q_k \rfloor$ is the number of bits required for storing quotient q_k, and (2.6) describes the constraint that the total allowed bit-storage for r quotients is equal to s. Let

$$n(s) := \min_{r} n(r, s). \quad (2.7)$$

Then for every integer s

$$q(s) = n(s) \quad (2.2). \quad (2.8)$$

3. Properties of Optimal Quotients

Consider $\{q_1, q_2, \cdots, q_{r-1}, q_r\}$ (2.3).

Then the following properties hold:

***Proposition* 3.1**: All optimal q_k must be exact powers of *two*.

Proof: Let's assume that the theorem is incorrect. This implies that for the same s, the value of $n(s)$ would be larger.

Indeed, consider for all $k \ge 1$

$$q_k = 2^{i_k} \le q'_k < 2^{i_k + 1}.$$

Then for all $k \ge 1$ the inequality $n_k \le n'_k$ holds. Here $n'_0 := b$; $n'_1 := a$ and all n'_k are generated itera- tively as $n'_k := q'_{k-1} n'_{k-1} + n'_{k-2}$. At the same time both arrays of quotients require the same size of bit storage.

Let $E_0 := I$ {identity matrix} and for all $i \ge 1$

$$E_i := \begin{pmatrix} 2^{i-1} & 1 \\ 1 & 0 \end{pmatrix}. \quad (3.1)$$

Then (2.5) may be rewritten as

$$n(r, s) := \min_{all\ q_k} (a, b) \prod_{k=1}^{r} E_{i_k} (1, 0)^T. \quad (3.2)$$

***Proposition* 3.2**: Since a spectral radius of matrix E_1^2 is larger than the spectral radius of matrix E_2, the sequence $\{n_1, \cdots, n_k, \cdots\}$, generated by an array of length $2m$ with all $q_k = 1$, grows faster than the sequence generated by an array of length m with all $q_k = 2$. Yet both arrays require the same storage. Hence all $q_k = 2$ generate smaller n_{r+1} than the *unary* array of the quotients. This observation provides a simple way to find an *upper bound* $h(s)$ for $n(s)$. Indeed, $h(0) = 2$, $h(2) = 5$, and for all $s \ge 2$

$$h(2s) = 2h(2s - 2) + h(2s - 4). \quad (3.3)$$

Remark 3.1: Let $H(s) := h(2s)$; then $H(0) = 2$;

$$H(1) = 5; H(s) := 2H(s - 1) + H(s - 2). \quad (3.4)$$

Representing the upper bound $H(s)$ as $H(s) = \alpha\rho^s + \beta\sigma^s$, [5,6], and using (3.4), we derive that

$$h(s) = \left(1 + 3\sqrt{2}\right)\left(\sqrt{1 + \sqrt{2}}\right)^s \Big/ 4 + o(s). \quad (3.5)$$

For $s = 40$, $h(40) = 93{,}222{,}358$, while the exact upper bound $n(40) = 80{,}198{,}051$ {see (8.3) below}, *i.e.*, the relative difference between $h(40)$ and $n(40)$ is more than 16%. However, for larger values of s this difference is significantly increasing, namely

$$\lim_{s \to \infty} h(s)/n(s) = \infty.$$

Let us now consider properties of control variables that help to determine their optimal values.

4. Diagonally Decreasing Matrices

4.1. Definition

$R=\left\{r_{ij}\right\}_{m\times n}$ is a diagonally decreasing matrix, {or *D*-matrix for short}, if $r_{ij}>r_{kl}$ for every $i\le k$ and $j\le l$. Hence for every $k\ge 1$, E_k are *D*-matrices.

4.2. Properties of D-Matrices

1) A product of *D*-matrices is a *D*-matrix;
2) Transposed *D*-matrix is also a *D*-matrix.

Let us consider the function

$$F\left(X,u,w\right):=\left(1,u\right)X\left(1,w\right)^T, \tag{4.1}$$

where $X\ge 0$ is a two-dimensional square matrix (all further inequalities involving matrices are to be taken entry-wise), $u>0$ and $w\ge 0$.

*Remark*4.1: For the sake of simplicity in forthcoming inequalities we use (wherever it is necessary) a normalization

$$\left(a,b\right)X\left(c,d\right)^T=ac\left(1,b/a\right)X\left(1,d/c\right)^T=acF\left(X,u,w\right), \tag{4.2}$$

where $u:=b/a$; $w:=d/c$.

Let
$$F\left(X\right):=F\left(X,.,.\right). \tag{4.3}$$

It is easy to verify that if $X\ge Y>0$, then

$$F\left(X\right)\ge F\left(Y\right). \tag{4.4}$$

For example, since $E_1^2\ge E_2$, then

$$F\left(E_1^2\right)\ge F\left(E_2\right). \tag{4.5}$$

5. Decomposition

***Proposition* 5.1**: Let

$$0<u<1;\quad 0\le w<1; \tag{5.1}$$

then the inequality

$$F\left(E_{k+m}\right)-F\left(E_kE_m\right)>0 \tag{5.2}$$

holds if

$$k\ge 1;\quad m\ge 1;\ k+m\ge 3,\ \text{and}\ \ w=0. \tag{5.3}$$

Proof: Consider

$$F\left(E_{k+m}\right)-F\left(E_kE_m\right)=\left(1,u\right)\left(E_{k+m}-E_kE_m\right)\left(1,w\right)^T>0, \tag{5.4}$$

and find under what conditions it holds.

Let $x:=2^{k-1}$; $y:=2^{m-1}$; then

$$\begin{aligned}E_{k+m}-E_kE_m&=\begin{pmatrix}2xy & 1\\ 1 & 0\end{pmatrix}-\begin{pmatrix}x & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}y & 1\\ 1 & 0\end{pmatrix}\\&=\begin{pmatrix}xy & 1\\ 1 & 0\end{pmatrix}-\begin{pmatrix}1 & x\\ y & 1\end{pmatrix}.\end{aligned} \tag{5.5}$$

Therefore, definitions (4.1) and (4.3), and Equations (5.4) and (5.5) imply the following inequality:

$$\left(2^{k-1}-u\right)\left(2^{m-1}-w\right)>\left(1-u\right)\left(1-w\right)+uw. \tag{5.6}$$

On the other hand, if (5.6) holds, then (4.4) also holds. In addition, it is sufficient to observe that (5.6) indeed holds if $k\ge 1$; $m\ge 1$; $k+m\ge 3$, and $w=0$.

However, (5.6) does *not* hold if $k=m=1$.

Q.E.D.

6. Transposition

***Proposition* 6.1**: The inequality

$$\left(1,u\right)\left(E_kE_m-E_mE_k\right)\left(1,w\right)^T>0 \tag{6.1}$$

holds if ($k>m$ and $w>u$)

or if ($k<m$ and $w<u$). (6.2)

Proof: Let $x:=2^{k-1}$; $y:=2^{m-1}$; $X:=\begin{pmatrix}x & 1\\ 1 & 0\end{pmatrix}$ and $Y:=\begin{pmatrix}y & 1\\ 1 & 0\end{pmatrix}$.

Thus,

$$XY-YX=\begin{pmatrix}0 & x-y\\ y-x & 0\end{pmatrix}=\left(x-y\right)\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}. \tag{6.3}$$

Therefore, inequality

$$\left(1,u\right)\left(XY-YX\right)\left(1,w\right)^T>0 \tag{6.4}$$

holds if
$$\left(x-y\right)\left(w-u\right)>0. \tag{6.5}$$

Hence, inequality (6.1) can be rewritten as

$$\left(2^{m-1}-2^{k-1}\right)\left(u-w\right)>0; \tag{6.6}$$

and (6.6) holds if

$$\operatorname{sign}\left(m-k\right)=\operatorname{sign}\left(u-w\right). \qquad \text{Q.E.D.}$$

In addition, inequality (6.6) implies that if $w=0$, then inequality (6.1) holds if $k<m$.

Let

$$E:=E_2E_1;\ E\left(1,v\right)^T=\lambda\left(1,v\right)^T \tag{6.7}$$

where λ is the largest eigenvalue of *E*.

Since $E=\begin{pmatrix}3 & 2\\ 1 & 1\end{pmatrix}>0$, *i.e.*, with all positive elements, then by Perron-Frobenius Theorem [7] its largest eigenvalue $\lambda>0$ with positive corresponding eigenvector $(1,v)$, where $v>0$ (6.7).

Indeed, $\lambda=2+\sqrt{3}$ and $v=\left(\sqrt{3}-1\right)/2$.

7. Optimal Control Variables

7.1. Cases *s* = 0, 1, 2

It is easy to verify that $n(0)=2; n(1)=3$. To find $n(2)$ we must compare two cases only. From the Decomposition Theorem it follows that $(a,b)E_1^2\geq(a,b)E_2$ for all $a>0$ and $b>0$.

Hence, $q_1^o=2$; and, if $(a,b)=(2,1)$, then $n(2)=5$.

7.2. Case *s* = 3

Let $X:=E_3$ and $E:=E_2E_1$.

Then

$$(2,1)E_3(1,0)^T>(2,1)E_1E_2(1,0)^T>(2,1)E_2E_1(1,0)^T \tag{7.1}$$

or after normalization

$$\begin{aligned}(1,1/2)E_3(1,0)^T&>(1,1/2)E_1E_2(1,0)^T\\&>(1,1/2)E_2E_1(1,0)^T\\&\equiv(1,1/2)E(1,0)^T\end{aligned} \tag{7.2}$$

Here $u=1/2$ and w = 0. Since in (7.2) w = 0, then the left-most inequality follows from the Decomposition {Proposition5.1} and the right-most inequality follows from the Transposition {Proposition6.1}. Thus, the optimal control variables are

$$q_1^o=2,\ \ q_2^o=1 \text{ and } n(3)=7.$$

7.3. Case *s* = 4

The following scheme shows that there are two local minima. Indeed, consider $X:=E_4$. Using the Decomposition and Transposition Theorems we can decrease the value of function $F(X)$. This procedure leads to two local minima: Let $A\Rightarrow B$ means that $F(A)\geq F(B)$, *i.e.*,

$$\begin{array}{ccccccccc}E_2E_1^2 & \Leftarrow & E_3E_1 & \Leftarrow & E_4 & \Leftarrow & E_1E_3 & \Leftarrow & E_1^2E_2\\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow\\ E_2^2 & & E_1E_2E_1 & & E_2^2 & & E_1E_2E_1 & & E_2^2\end{array}$$

Direct comparison implies that

$$F\left(E_2^2\right)>F\left(E_1E_2E_1\right)\equiv F\left(E_1E\right). \tag{7.3}$$

Hence the optimal control variables are equal

$q_1^0=q_3^0=1$ and $q_2^0=2$.

Thus $n(4)=11$.

7.4. Case *s* = 5

Consider $X:=E_5$. Systematically applying decomposition and transposition, it is possible to demonstrate, that there are two local minima: $E_2E_1E_2$ and E_2E only (6.7). {see the diagram below}. The direct comparison shows that E_2E delivers the global minimum.

***Proposition* 7.1:** Let p_0 and p_1 be a pair that requires *s* bits of storage; $L:=(2,1)$; $R:=(1,0)^T$; $s=3m+j$; $0\leq j\leq 2$ and

$$\sum_{\{all\ k\}} i_k=s. \tag{7.4}$$

Then

$$\min_{\{all\ q\}} L\prod_i{}_{qi}E_{q_i}R=LE_jE^mR. \tag{7.5}$$

Proof: (by induction over *m*).

(1) *m* = 0: for *j* = 1 and *j* = 2 we proved in the Subsection 6.1 that LE_jR is the minimum.

For *j* = 3 we proved in (6.1)-(6.2) that $L(E_3-E_1E_2)R>0$, and that $L(E_1E_2-E_2E_1)E^mR>0$, *i.e.*, LER is the minimum.

(2) Let for $m=0,1,\cdots,k$ LE_jE^mR be the minimum if *j* = 0, 1, 2; { $E_0:=I$, (3.1)} and correspondingly E_jE^m be the optimal control strategy.

(3) Let us insert matrix E_3 into LE_jE^kR and prove that the following two inequalities hold:

$$LE_jE^k(E_3-E_1E_2)R>0 \tag{7.6}$$

and

$$LE_jE^k(E_1E_2-E_2E_1)R>0. \tag{7.7}$$

Let $LE_j:=(a_{0j},b_{0j})$, and for all *m*

$$(a_{mj},b_{mj}):=(a_{0j},b_{0j})E^m. \tag{7.8}$$

Here L, E_j, E and R are *D*-matrices, {see the Definition 3.1}. Hence LE_j and LE_jE^k are also *D*-matrices, {as products of *D*-matrices}, *i.e.*, $a_{ij}>b_{ij}>0$ for all *i* = 0, 1, 2,⋯

Dividing the inequalities (7.6) and (7.7) by a_{kj}, we can respectively rewrite them as

$$(1,u_{kj})(E_3-E_1E_2)(1,0)^T>0 \tag{7.9}$$

and as

$$\left(1,u_{kj}\right)\left(E_1E_2-E_2E_1\right)\left(1,0\right)^T>0 \qquad (7.10)$$

Then inequality (7.9) holds by the Decomposition Theorem because $w = 0$, $0<u_{kj}<1$, and $u_{kj}:=b_{kj}/a_{kj}$. On the other hand, inequality (6.10) holds by the Transposition Theorem because $u_{kj}>w=0$. Therefore $LE_jE^k(E_2E_1)R=LE_jE^kER=LE_jE^{k+1}R$ is the minimum and E_jE^{k+1} is the optimal control strategy.

Q.E.D.

***Remark* 7.1**: Another (more tedious) way to prove the Theorem is to consider an induction over *m* and *j*. Firstly, we make an assumption that for all $m=0,1,\cdots,k$ and for all *j* = 0, 1, 2 LE_jE^mR is the minimum and E_jE^m is the optimal strategy. Then we prove that for every

$1\le i\le m$

$$F\left(E_jE^iE_1E^{m-i}\right)>F\left(E_jE^{i-1}E_1E^{m-i+1}\right)>\cdots>F\left(AE^m\right). \qquad (7.11)$$

Here $E_0:=I$; $A:=E_{j+1}$ if *j* = 0 or *j* = 1; and $A:=E$ if *j* = 2. The application of all transpositions is based on the following propositions (provided here without proofs):

***Proposition* 7.2**: for all *j* = 0, 1, 2 and for all $i=1,2,\cdots$

$$u_{0j}>u_{1j}>\cdots>u_{ij}>\cdots>v>\cdots>w_l>\cdots>w_1>w_0=0, \qquad (7.12)$$

if

$$u_{0j}>v=\left(\sqrt{3}-1\right)/2. \qquad (7.13)$$

Here all u_{ij} are defined in (4.2) and (7.8), and *v* satisfies the condition $(1,v)E^T=\lambda(1,v)$, *i.e.*, *v* is the second component of a normalized eigenvector of matrix *E*, corresponding to the largest eigenvalue λ of *E*, [7]. Direct computation shows that indeed the inequalities $u_{0j}>v$ are satisfied for every *j* = 0, 1, 2.

Therefore,

$$F\left(E_jE^iE_1E^{m-i}\right)>F\left(E_jE^{i-1}E_1E^{m-i+1}\right) \qquad (7.14)$$

holds for every $i=1,\cdots,m$.

8. Optimal Control Variables

Let $n(s)$ be a minimal p_0 that requires no more than *s* bits of storage for the stack. The minimal values $n(s)$ are generated by the following optimal quotients q_k^0:

1) If *s* = 3*m*, then for every $k\ge1$

$$q_k^0=1+k \bmod 2; \qquad (8.1)$$

2) If *s* = 3*m* + *r*, and *r* = 1,2, then $q_1^0=r$, and for every $k\ge2$

$$q_k^0=2-k \bmod 2. \qquad (8.2)$$

Examples: $n(0)=2$; $n(1)=3$; $n(2)=5$; $n(3)=7$; $n(4)=11$; $n(5)=17$; $n(6)=26$; $n(7)=41$; $n(8)=63$; $n(9)=97$; $n(10)=153$; $n(11)=235$; $n(12)=362$; $n(13)=571$; $n(14)=877$; $n(15)=1412$; $n(20)=12,863;\cdots;\ n(40)=80,198,051.$ (8.3)

9. Telescopic Relations for Tight Upper Bound $n(s)$

Since for *i* = 0, 1, 2

$$n(3m+i)=(2,1)E_iE^m(1,0)^T, \qquad (9.1)$$

let us find telescopic relations for $n(s)$ in the following form [7]:

$$n(3m+i+2)=x_in(3m+i+1)+y_in(3m+1) \qquad (9.2)$$

where x_i and y_i must satisfy equations

$$(2,1)\left(A_{i+2}-x_iA_{i+1}-y_iA_i\right)E_2E^m(1,0)^T=0; \qquad (9.3)$$

and where

$$A_i=\begin{cases}E_i, i=0,1,2\\ E, \quad i=3\\ E_1E, \ i=4\end{cases}. \qquad (9.4)$$

From these equations we find all x_i and y_i and establish the following telescopic relations for $n(s)$:

$$4n(3m)=-n(3m-1)+11n(3m-2); \qquad (9.5)$$

$$11n(3m+1)=18n(3m)-n(3m-1); \qquad (9.6)$$

$$n(3m+2)=-n(3m+1)+4n(3m) \quad [6]. \qquad (9.7)$$

Therefore

$$n(3m-2)=\left[n(3m)+n(3m-3)\right]/3; \qquad (9.8)$$

$$n(3m-1)=\left[-n(3m)+11n(3m-3)\right]/3; \qquad (9.9)$$

and finally, $n(3m)=\left[n(3m+3)+n(3m-3)\right]/4$ or

$$n(3m+3)=4n(3m)-n(3m-3). \qquad (9.10)$$

10. Closed-Form Expressions for $n(s)$

Direct substitution shows that

$$n(3m)=\left[\left(2+\sqrt{3}\right)^{m+1}+\left(2-\sqrt{3}\right)^{m+1}\right]/2. \qquad (10.1)$$

satisfies (9.10) for every integer $m\ge0$.

Using (9.8), (9.9) and (10.1), we can find closed-form expressions for

$$n(3m-2) = \left[\left(2+\sqrt{3}\right)^m\left(1+1/\sqrt{3}\right)+\left(2-\sqrt{3}\right)^m\left(1-1/\sqrt{3}\right)\right]/2 \quad (10.2)$$

and for

$$n(3m-1) = \left[\left(2+\sqrt{3}\right)^m\left(3-1/\sqrt{3}\right)+\left(2-\sqrt{3}\right)^m\left(3+1/\sqrt{3}\right)\right]/2 \quad (10.3)$$

The tight-upper bound on the required bit-storage is deducible from the following

***Proposition* 10.1**: The bit-storage required by EEA for storage of the stack in the worst case satisfies equation:

$$\max_{2\le p_1<p_0} N(p_0,p_1) = 3\left(\log_{2+\sqrt{3}} p_0\right)\left[1+o(p_0)\right], \quad (10.4)$$

Proof: Since $\left|2-\sqrt{3}\right|<1$, then definitions (2.1)-(2.2) and formulas (10.1)-(10.3) imply (10.4).

11. Asymptotic Rate of Growth/Bit

Let $s=3m+i$; represent $n(s)$ in a form

$$n(3m+i) = a_i u^{3m+i} + o(m) \quad (11.1)$$

The asymptotic rate of growth u equals to $r(E)$, where $r(E)$ is a spectral radius of matrix E, [8-10], *i.e.*, the largest root of equation $\begin{vmatrix} 3-\lambda & 2 \\ 1 & 1-\lambda \end{vmatrix} = 0$.

Since $\lambda_{1,2} = 2\pm\sqrt{3}$, then

$$u = \sqrt[3]{2+\sqrt{3}} = 1.5511335 \quad (11.2)$$

This observation independently verifies the stronger results of (10.1)-(10.3). Indeed, we can find the asymptotic rate of growth/bit by taking into account that $\left(2-\sqrt{3}\right)^m \to 0$ in (10.1)-(10.3) for large m. Finally, since $\sqrt{1+\sqrt{2}} > \sqrt[3]{2+\sqrt{3}}$, then the tight upper bound $n(s)$ grows slower than $h(s)$ (3.5).

Main Theorem: Let $N(p_0)$ denote the maximal number of bits required to store all quotients in the stack if the modulus equals p_0; and let $u := \sqrt[3]{2+\sqrt{3}}$. If

$$p_0 \in \left[a_0 u^{3m}, a_1 u^{3m+1}-1\right], \quad \text{then } N(p_0) \le 3m;$$

$$p_0 \in \left[a_1 u^{3m+1}, a_2 u^{3m+2}-1\right], \text{ then } N(p_0) \le 3m+1;$$

$$p_0 \in \left[a_2 u^{3m+2}, a_0 u^{3(m+1)}-1\right], \text{ then } N(p_0) \le 3m+2 \quad (11.3)$$

where $a_0 = 1/2$; $a_1 = \left(3-1/\sqrt{3}\right)/2$; and $a_2 = \left(1+1/\sqrt{3}\right)/2$.

Proof follows from (11.1) and (10.1)-(10.4).

12. Conclusions

As it follows from the observed results, Enhanced-Euclid algorithm requires very small bit-storage for its execution. This storage does not exceed a 2K-bit level for public-key encryption algorithms, dealing with numbers p_0 and p_1 of range $(10^{100}, 10^{400})$. As it is demonstrated in numerous computer experiments, the average bit-storage is actually 40% *smaller* than 2 K. Hence the EEA is executable if necessary by a custom-built chip with small memory, [11]. This property of the Enhanced-Euclid algorithm is especially important for a potential implementation of encryption if integrated-circuit memory is limited, (smart cards, PC cards, cell phones, wearable computers etc.).

In the analysis provided above, it is not considered storage space, required for delimiters separating the quotients in the stack. One way to resolve the retrieval problem is to use a dynamic *prefix* coding for the quotients [12,13]. Since in the prefix coding there are no two codes such that one code is a prefix of another code, the quotients can be retrieved from the stack without delimiters.

13. Acknowledgements

I express my appreciation to R. Rubino, D. Murphy and to anonymous reviewers for suggestions that improved the style of this paper.

14. References

[1] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography," CRC Press, Boca Raton, 1997.

[2] D. E. Knuth, "Fundamental Algorithms," 3rd Edition, *The Art of Computer Programming*, Vol. 1, Addison-Wesley, Reading, MA, 1997.

[3] B. Verkhovsky, "Enhanced Euclid Algorithm for Modular Multiplicative Inverse and Its Complexity," *Proceedings of* 10*th International Conference on Systems Research, Informatics and Cybernetics*, Baden-Baden, 17-22 August 1998.

[4] B. Verkhovsky, "Enhanced Euclid Algorithm for Modular Multiplicative Inverse and Its Cryptographic Application," *International Journal of Communications, Network and System Sciences*, Vol. 3, No. 12, 2010, pp. 901-906. doi:10.4236/ijcns.2010.312123

[5] D. Harkin, "On the Mathematical Works of Francois-Edouard-Anatole Lucas," *Enseignement Mathematique*, Vol. 3, No. 2, 1957, pp. 276-288.

[6] G. S. Lueker, "Some Techniques for Solving Recurrences," *ACM Computing Surveys*, Vol. 12, No. 4, 1980, pp. 410-436. doi:10.1145/356827.356832

[7] O. Perron, "Zur Theorie der Matrizen," *Mathematische Annalen*, in German, Vol. 64, 1907, pp. 248-263.

[8] J. Hartmanis and R. E. Stearns, "On the Computational Complexity of Algorithms," *Transactions of the American Mathematical Society*, Vol. 117, No. 5, 1965, pp. 285-306. doi:10.1090/S0002-9947-1965-0170805-7

[9] L. Fortnow and S. Homer, "A Short History of Computational Complexity," *Bulletin of the European Association for Theoretical Computer Science*, Vol. 80, 2003, pp. 95-133.

[10] O. Goldreich, "Computational Complexity: A Conceptual Perspective," Cambridge University Press, Cambridge, 2008.

[11] P. Ivey, S. N. Walker, J. M. Stern and S. Davidson, "An Ultra-High Speed Public Key Encryption Processor," *Proceeding of IEEE Custom Integrated Circuits Conference*, Boston, 3-6 May 1992, pp. 19.6.1-19.6.4. doi:10.1109/CICC.1992.591336

[12] J. S. Vitter, "Design and Analysis of Dynamic Huffman Codes," *Journal of the ACM*, Vol. 34, No. 4, 1987, pp. 825-845. doi:10.1145/31846.42227

[13] J. S. Vitter, "ALGORITHM 673: Dynamic Huffman Coding," *ACM Transactions on Mathematical Software*, Vol. 15, No. 2, 1989, pp. 158-167. doi:10.1145/63522.214390

Appendix

$$E_5 \Rightarrow E_1E_4 \Rightarrow E_1E_3E_1 \Rightarrow E_1^2E_2E_1 \Rightarrow E_2^2E_1\,; \qquad \text{(A.1)}$$

$$E_5 \Rightarrow E_3E_2 \Rightarrow E_1E_2^2 \Rightarrow E_2E_1E_2 \Rightarrow E_2^2E_1; \qquad \text{(A.2)}$$

$$E_5 \Rightarrow E_2E_3 \Rightarrow E_2^2E_1 \qquad \text{(A.3)}$$

Diagrams (A.1)-(A.3) show that $E_2^2E_1$ delivers the global minimum.

