Fişa suspiciunii de plagiat / Sheet of plagiarism's suspicion	Indexat la: 29/06

Opera suspicionată (OS)	Opera autentică (OA)
Suspicious work	Authentic work

OS	Mang, I., Mang, E., Public-key cryptosystem based on the diophantine equations, *Journal of Computer Science and Control Systems*, Vol.1 Nr.1, p.58-61, 2008. Disponibil la: http :// electroinf . uoradea . ro / reviste %20 CSCS/documente / JCSCS_2008 / JCSCS _ 2008 _ 11 _ MangI _ 1 . pdf.
OA	Lin, C.H. Chang, C.C. Lee, R.C.T., „A New Public-Key Cipher System Based Upon the Diophantine Equations", *IEEE Trans. on Computers*, vol. 44, no. 1, p. 13-19, Jan. 1995.

Incidența minimă a suspiciunii / Minimum incidence of suspicion	
p.58: 9s – p.60:18d	p.13 - 12d - p.15:47s

Fişa întocmită pentru includerea suspiciunii în Indexul Operelor Plagiate în România de la www.plagiate.ro

A New Public-Key Cipher System Based Upon the Diophantine Equations

C. H. Lin, C. C. Chang, *Senior Member, IEEE,* and R. C. T. Lee, *Fellow, IEEE*

***Abstract*— A new public-key (two-key) cipher scheme is proposed in this paper. In our scheme, keys can be easily generated. In addition, both encryption and decryption procedures are simple. To encrypt a message, the sender needs to conduct a vector product of the message being sent and the enciphering key. On the other hand, the receiver can easily decrypt it by conducting several multiplication operations and modulus operations. For security analysis, we also examine some possible attacks on the presented scheme.**

***Index Terms*— Public keys, private keys, cryptosystems, Diophantine equation problems, integer knapsack problems, one-way functions, trapdoor one-way functions, NP-complete.**

I. Introduction

In [6], Diffie and Hellman proposed their pioneering idea of public key cryptosystems. In a public key system, each user U uses the encryption algorithm $E(PK_u, M)$ and the decryption algorithm $D(PR_u, C)$, where PK_u is the public key, PR_u is the private key of U and M and C are the texts to be encrypted or to be decrypted, respectively. Each user publishes his encryption key by putting it on a public directory, while the decryption key is kept secret by himself. Suppose that user A wants to send a message M to user B. First, A finds the public encryption key, namely PK_b, for B from the public directory. Then A encrypts the message M to C by $C = E(PK_b, M)$ and sends C to B. On receiving C, B can decode it by computing $M = D(PR_b, C)$. Since PR_b is private for B, no one else can perform this decryption process. Therefore, for practical purposes, the encryption and decryption algorithms E and D have to satisfy the following three requirements.

1) $D(PR_u, E(PK_u, M)) = M$
2) Neither of algorithms E and D needs much computing time.
3) To derive the associate PR_u from the publicly known PK_u is computationally infeasible [5].

A number of public-key cryptosystems have been proposed [1], [3], [7], [9], [17], [20]–[22], [26]. These systems can be put into two categories. One is based on hard number theoretic problems such as factoring, taking discrete logarithms, etc.; while the other is related to NP-complete problems such as 0/1 knapsack and so on. To construct cryptosystems based on these computationally hard problems, secret "trapdoor" information is added such that a one-way function is invertible. A function F is called a one-way function if and only if the computation of $F(x)$ is easy for all x in the domain of F, while it is computationally infeasible to compute the inverse $F^{-1}(y)$ given any y in the range of F, even if F is known. It is a trapdoor one-way function if the inverse becomes easy when certain additional information is given. This additional information is used as a secret decryption key.

In this paper, a new public-key cipher scheme is proposed. By the use of our scheme, the generating steps of keys are simple. Both the encryption and decryption procedures can be completed efficiently. Our cipher scheme is based upon the Diophantine equations [18]. In general, a Diophantine equation is defined as follows: We are given a polynomial equation $f(x_1, x_2, \cdots, x_n) = 0$ with integer coefficients and we are asked to find rational or integral solutions. Throughout this paper, we shall assume that the solutions are nonnegative. For instance, consider the following equation:

$$3x_1 + 4x_2 + 7x_3 + 5x_4 = 78.$$

The above equation is a Diophantine equation if we have to find a nonnegative solution for this equation. In fact, our solution is $(x_1, x_2, x_3, x_4) = (2, 5, 1, 9)$. Another example of a Diophantine equation is

$$3x_1^3 x_2 + 4x_1 x_2 x_3 + 5x_4 = 105.$$

Diophantine equations are usually hard solve. In [14], it was proved that the problem of deciding whether there are positive integer solutions for

$$\alpha x_1^2 + \beta x_2 - \gamma = 0,$$

where α, β and γ are positive integers, is NP-complete [4], [8]. Some specific cases of Diophantine equations and their computational complexities were studied in [24], [25].

A famous Diophantine equation problem is Hilbert's tenth problem [11], which is defined as follows: Given a system of polynomials $P_i(x_1, x_2, \cdots, x_n)$, $1 \leq i \leq m$, with integer coefficients, determine whether it has a nonnegative integer solution or not. In [15] and [23], it was shown that the Hilbert problem is undecidable for polynomials with degree 4. It was shown in [16] that the Hilbert problem is undecidable for polynomials with 13 variables. Gurari and Ibarra [10] also proved that several Diophantine equations are in NP-complete class.

Manuscript received May 7, 1991; revised July 19, 1992.

C. H. Lin is with the Department of Computer and Information Sciences, Tunghai University, Taichung, Taiwan 40704, R.O.C.
C. C. Chang is with the Institute of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan 62107, R.O.C.
R. C. T. Lee is with the Department of Computer Sciences, Providence University, Shalu, Taichung Hsien, Taiwan 43301, R.O.C.; E-mail: rctlee @ host1.pu.edu.tw.

IEEE Log Number 9407115.

0018-9340/95$04.00 © 1995 IEEE

Finally, we sketch the organization of this paper as follows. Underlying mathematics is described in Section II. The generation of the system, encryption and decryption algorithms, will appear in Section III. Section IV investigates the security of our cipher scheme. We also show that in order to break our system, one has to solve some specific Diophantine equations. Finally, conclusions are made in Section V.

II. The Underlying Mathematics

In this section, we describe the mathematics on which the new cryptosystem is based. Let w be some positive integer and the domain $\mathcal{D}$ be a set of positive integers in the range of $[0, w]$. Let $w = 2^b - 1$, where b is some positive integer. Assume that a sending message M with length nb bits is broken up into n pieces of submessages, namely $m_1, m_2, \cdots$ and m_n. Each submessage is of length b bits. In other words, we can represent each submessage by a decimal number m_i and m_i in $\mathcal{D}$.

Suppose that n pairs of integers $(q_1, k_1), (q_2, k_2), \cdots$, and $(q_n, k_n)$are chosen such that the following conditions hold:

1) q_i's are pairwise relative primes; i.e., $(q_i, q_j) = 1$ for $i \neq j$.
2) $k_i > w$ for $i = 1, 2, \cdots, n$.
3) $q_i > k_i w(q_i \bmod k_i)$, and $q_i \bmod k_i \neq 0$, for $i = 1, 2, \cdots, n$.

These n integer pairs (q_i, k_i)'s will be kept secret and used to decrypt messages. For convenience, we name the above three conditions the DK-conditions since they will be used as deciphering keys. Note that for the generating of pairwise relatively primes, one can consult [2]. Furthermore, the following numbers are computed. First, compute $R_i = q_i \bmod k_i$ and compute P_i's such that two conditions are satisfied: 1) $P_i \bmod q_i = R_i$, and 2) $P_j \bmod q_i = 0$ if $i \neq j$. Since q_i's are pairwise relatively primes, one solution for P_i's satisfying the above two conditions is that $P_i = Q_i b_i$ with

$$Q_i = \prod_{j \neq i} q_i$$

and b_i is chosen such that $Q_i b_i \bmod q_i = R_i$. Since Q_i and q_i are relatively prime, b_i's can be found by using the extended Euclid's algorithm [5]. Note that the average number of divisions performed by the extended Euclid's algorithm for finding b_i is approximately $0.843 \cdot \ln(q_i) + 1.47$ [13]. Secondly, compute $N_i = \lceil q_i/(k_i R_i) \rceil$ for $i = 1, 2, \cdots, n$. Finally, compute

$$s_i = P_i N_i \bmod Q, \quad \text{where } Q = \prod_{i=1}^{n} q_j. \tag{1}$$

That is, we have a vector $S = (s_1, s_2, \cdots, s_n)$ with each component computed as above.

After this, S can be used as the enciphering key for encrypting messages. By conducting a vector product between $M = (m_1, m_2, \cdots, m_n)$ and $S = (s_1, s_2, \cdots, s_n)$; i.e.,

$$C = E(S, M) = M * S = \sum_{i=1}^{n} m_i s_i \tag{2}$$

a message M is transformed to its ciphertext C, where $*$ denotes the vector product operation. Conversely, the ith component m_i in M can be revealed by the following operation:

$$m_i = D((q_i, k_i), C) = \lfloor k_i C/q_i \rfloor \bmod k_i \quad \text{for } i = 1, 2, \cdots, n. \tag{3}$$

Theorem 2.1 shows that (3) is the inverse function of (2). The following lemmas are helpful in the proof of the theorem.

Lemma 2.1: Let a and b be some positive integers where $b > a$. Then for all x, $a\lceil x/b \rceil < x$ if $x \geq ab/(b-a)$.

Proof: Let $\lceil x/b \rceil = c$ for some integer c. Then $x/b \leq c < (x/b + 1)$. We have

$$ac < (ax/b + a). \tag{4}$$

On the other hand, if $x \geq ab/(b-a)$, then $(b-a)x \geq ab$; that is,

$$(ax/b + a) \leq x. \tag{5}$$

Combining (4) and (5), we have that $a\lceil x/b \rceil < x$ if $x \geq ab/(b-a)$. □

Lemma 2.2: Let $R_i = q_i \bmod k_i$. Then $k_i R_i m_i \lceil q_i/(k_i R_i) \rceil \bmod k_i q_i = k_i R_i m_i \lceil q_i/(k_i R_i) \rceil$.

Proof: Let $a = R_i m_i$, $b = k_i R_i$, and $x = q_i$. Since $q_i > k_i R_i w$, we know that $q_i > k_i R_i^2 m_i/(R_i(k_i - m_i))$. That is, $x \geq ab/(b-a)$ is satisfied. By applying Lemma 2.1, it can be seen that $R_i m_i \lceil q_i/(k_i R_i) \rceil < q_i$. Therefore, $k_i R_i m_i \lceil q_i/(k_i R_i) \rceil \bmod k_i q_i = k_i R_i m_i \lceil q_i/(k_i R_i) \rceil$. □

Lemma 2.3: Let m_i's, k_i's, and q_i's be chosen such that the DK-conditions are satisfied. Let $R_i = q_i \bmod k_i$. Then $\lfloor k_i R_i m_i \lceil q_i/(k_i R_i) \rceil / q_i \rfloor = m_i$.

Proof: Let $\delta = \lfloor k_i R_i m_i \lceil q_i/(k_i R_i) \rceil / q_i \rfloor$. It can be easily seen that the following two inequalities hold:

$$\delta < \lfloor k_i R_i m_i (q_i/(k_i R_i) + 1)/q_i \rfloor \tag{6}$$

and

$$\delta \geq \lfloor k_i R_i m_i (q_i/(k_i R_i))/q_i \rfloor. \tag{7}$$

Furthermore, the right-hand side of (7) is identical to m_i and that of (6) is $\lfloor m_i + k_i R_i m_i/q_i \rfloor$. On the other hand, since m_i is an integer and $k_i R_i m_i/q_i < 1$, the right-hand side in (6) becomes $\lceil m_i + k_i R_i m_i/q_i \rceil = m_i$. Combining these two inequalities, we obtain that $m_i \leq \delta < m_i$. Finally, we have $\delta = m_i$, since δ is an integer. □

Theorem 2.1: Let $(q_1, k_1), (q_2, k_2), \cdots$, and (q_n, k_n) be n pairs of positive integers satisfying the DK-conditions. Let the vector S be computed by applying (1). Then (3) is the inverse function of (2). That is, a message enciphered by (2) can be decrypted by (3).

Proof: Let us prove the theorem by the following two steps. First, from (1), define $\bar{s}_i = P_i N_i$; we have a vector $\bar{S} = (\bar{s}_1, \bar{s}_2, \cdots, \bar{s}_n)$; i.e., $s_i = \bar{s}_i \bmod Q$, for $i = 1, 2, \cdots, n$. Let $C' = M * \bar{S} = \sum_{i=1}^{n} m_i \bar{s}_i = \sum_{i=1}^{n} m_i P_i N_i$. Since P_i's satisfy the following two conditions, 1) $P_i \bmod q_i = q_i \bmod k_i = R_i$; and 2) $P_j \bmod q_i = 0$ if $i \neq j$, $k_i C' \bmod$

$k_iq_i = (k_i \sum_{i=1}^{n} m_iP_iN_i) \bmod k_iq_i = k_im_iR_i\lceil q_i/(k_iR_i)\rceil \bmod k_iq_i$. Furthermore, by Lemma 2.2, $k_im_iR_i\lceil q_i/(k_iR_i)\rceil \bmod k_iq_i = k_im_iR_i\lceil q_i/(k_iR_i)\rceil$. That is, $k_iC' \bmod k_iq_i = k_im_iR_i\lceil q_i/(k_iR_i)\rceil$ for $i = 1, 2, \cdots, n$. In other words, $k_iC' = y_ik_iq_i + k_im_iR_i\lceil q_i/(k_iR_i)\rceil$ for some integers y_i. Moreover, $k_iC'/q_i = y_ik_i + k_im_iR_i\lceil q_i/(k_iR_i)\rceil/q_i$. Hence $\lfloor k_iC'/q_i\rfloor = \lfloor y_ik_i + k_im_iR_i\lceil q_i/(k_iR_i)\rceil/q_i\rfloor = y_ik_i + \lfloor k_im_iR_i\lceil q_i/(k_iR_i)\rceil/q_i\rfloor$. By applying Lemma 2.3, we have $\lfloor k_iC'/q_i\rfloor = y_ik_i + m_i$. Thus $m_i = \lfloor k_iC'/q_i\rfloor \bmod k_i$.

Second, let

$$Q = \prod_{i=1}^{n} q_i$$

then $C' \bmod Q = (\sum_{i=1}^{n} m_i\bar{s}_i) \bmod Q = ((m_1\bar{s}_1 \bmod Q) + \cdots + (m_n\bar{s}_n \bmod Q)) \bmod Q = (m_1(\bar{s}_1 \bmod Q) + \cdots + m_n(\bar{s}_n \bmod Q)) \bmod Q = (\sum_{i=1}^{n} m_is_i) \bmod Q = C \bmod Q$. That is, $C' \equiv C (\bmod Q)$. Let $C' = C + zQ$, for some positive integer z. We have $\lfloor k_iC/q_i\rfloor \bmod k_i = (\lfloor k_i(C' - zQ)/q_i\rfloor \bmod k_i = (\lfloor k_iC'/q_i\rfloor - k_izQ_i) \bmod k_i = \lfloor k_iC'/q_i\rfloor \bmod k_i$. In other words, $m_i = \lfloor k_iC/q_i\rfloor \bmod k_i$. □

III. The Construction and Usage of the Cryptosystem

In this section, how the new cryptosystem is created and used is described. First, an informal description is given. Then algorithms for constructing the cryptosystem, encrypting messages, and decrypting messages, respectively, are presented.

First, each user picks n pairs of parameters (q_1, k_1), (q_2, k_2), $\cdots$, and (q_n, k_n) such that the DK-conditions are satisfied. Afterward,

$$Q_i = \prod_{j \neq i} q_j$$

and $N_i = \lceil q_i/(k_i(q_i \bmod k_i))\rceil$ are computed, and b_i's are integers chosen such that $Q_ib_i \bmod q_i = q_i \bmod k_i$, for $i = 1, 2, \cdots, n$. Let $P_i = Q_ib_i$ and $s_i = P_iN_i \bmod Q$, for $i = 1, 2, \cdots, n$, where

$$Q = \prod_{i=1}^{n} q_i.$$

Therefore, a vector $S = (s_1, s_2, \cdots, s_n)$ is obtained. Then the n-tuple S of integers is published and used as the public key of the cryptosystem for enciphering messages.

The chosen parameters (q_1, k_1), (q_2, k_2), $\cdots$, and (q_n, k_n) are kept and used as the private key to decipher messages received. Specifically, let user A be the sender and user B be the receiver, and let A be sending a message represented by

$$M = (m_1, m_2, \cdots, m_n),$$

where m_i is a b-bits submessage represented by a decimal number in the range of $[0, 2^b - 1]$. Then $(m_1, m_2, \cdots, m_n)$ is enciphered by (2) into an integer C. Afterward, the integer C is sent to user B as the ciphertext of the original message M. On the receiving of integer C, user B is able to convert C into $(m_1, m_2, \cdots, m_n)$ by applying (3).

Algorithm 3.1—Key Generating for Each User U:

Step 1. Pick n pairs of positive integers (q_1, k_1), (q_2, k_2), $\cdots$, and (q_n, k_n) such that the DK-conditions are satisfied.

Step 2. Compute $R_i = q_i \bmod k_i$ for $i = 1, 2, \cdots, n$. Compute

$$Q_i = \prod_{j \neq i} q_i$$

and $N_i = \lceil q_i/(k_iR_i)\rceil$, for $i = 1, 2, \cdots, n$, and compute

$$Q = \prod_{i=1}^{n} q_i.$$

Step 3. Compute b_i's such that $Q_ib_i \bmod q_i = R_i$ for $i = 1, 2, \cdots, n$. This can be done by the extended version of Euclid's algorithm.

Step 4. Compute $P_i = Q_ib_i$ and $s_i = P_iN_i \bmod Q$ for $i = 1, 2, \cdots, n$.

Step 5. Publish the encryption key $PK_u = (s_1, s_2, \cdots, s_n)$ for user U.

Step 6. Keep the private decryption key $PR_u = ((q_1, k_1), (q_2, k_2), \cdots, (q_n, k_n))$ in secret.

Step 7. Keep P_i, Q_i, b_i, N_i, and Q in secret or erase them.

Algorithm 3.2—Encryption Procedure for Sender A:

Step 1. Encrypt $M = (m_1, m_2, \cdots, m_n)$ by (2); i.e., $C = E(S, M) = S * M$.

Step 2. Send out the integer C as the ciphertext of message M.

Step 3. Exit.

Algorithm 3.3—Decryption Procedure for Receiver B:

Step 1. Compute the ith component m_i of message M by computing $m_i = D((q_i, k_i), C) = \lfloor k_iC/q_i\rfloor \bmod k_i$, $1 \leq i \leq n$.

Step 2. Exit.

In the following, let us illustrate the processing of the presented cipher scheme by a simple example.

Example 3.1: Consider a simple case with $n = 3$. Let $(q_1, k_1) = (104, 6)$, $(q_2, k_2) = (147, 8)$, and $(q_3, k_3) = (121, 7)$. Then $R_1 = q_1 \bmod k_1 = 2$, $R_2 = q_2 \bmod k_2 = 3$, and $R_3 = q_3 \bmod k_3 = 2$. Let $\mathcal{D} = \{0, 1, 2, 3\}$ with $w = 3$. It can be verified that the DK-conditions are satisfied in this case.

Since $Q_1 = 17787$, $Q_2 = 12584$, and $Q_3 = 15288$, and $Q = 1849848$, if $b_1 = 70$, $b_2 = 114$, and $b_3 = 98$ are chosen, we have $P_1 = Q_1b_1 = 1245090$, and $P_2 = Q_2b_2 = 1434576$, $P_3 = Q_3b_3 = 1498224$. Moreover, since $N_1 = \lceil q_1/(k_1R_1)\rceil = 9$, $N_2 = \lceil q_2/(k_2R_2)\rceil = 7$, and $N_3 = \lceil q_3/(k_3R_3)\rceil = 9$, we have $s_1 = P_1N_1 \bmod Q = 106722$, $s_2 = P_2N_2 \bmod Q = 792792$, and $s_3 = P_3N_3 \bmod Q = 535080$. In other words, a vector $S = (106722, 792792, 535080)$ is obtained.

Now, we assume that user A wants to send a message M, say represented by binary string 111101. Let M be broken up into three submessages with length 2-bit; i.e., $M = (11, 11, 01)$ or $M = (m_1, m_2, m_3) = (3, 3, 1)$ in decimal representation. A also computes $C = (m_1, m_2, m_3) *$

mod Q and P_i mod $q_i = R_i$, he can deduce that $s_i \equiv R_i N_i \pmod{q_i}$ for $i = 1, 2, \cdots, n$. In other words, the following equations are obtained:

$$s_i \bmod q_i = R_i N_i = R_i \lceil q_i/(k_i R_i) \rceil$$

where

$$R_i = q_i \bmod k_i,\ 1 \le i \le n. \tag{8}$$

Equation (8) can be rewritten as

$$s_i = q_i x_i + R_i \lceil q_i/(k_i R_i) \rceil, \text{ for some } x_i,\ 1 \le i \le n. \tag{9}$$

Let $v_i = \lceil q_i/(k_i R_i) \rceil$. Then $v_i - 1 < q_i/(k_i R_i) \le v_i$ and $k_i R_i (v_i - 1) < q_i \le k_i R_i v_i$. We have

$$q_i = k_i R_i (v_i - 1) + y_i, \text{ with } 1 \le y_i \le k_i R_i,\ 1 \le i \le n. \tag{10}$$

Substituting (10) into (9), we obtain the following equations

$$k_i R_i (v_i - 1) x_i + y_i x_i + R_i v_i - s_i = 0$$

with

$$1 \le y_i \le k_i R_i,\ 1 \le i \le n. \tag{11}$$

Equation (11) is a system of n Diophantine equations with degree 4 and has variables k_i, R_i, v_i, x_i, and y_i, for $1 \le i \le n$. Our job of breaking the cipher system consists of the following steps:

Step 1. Find k_i, R_i, v_i, x_i, and y_i satisfying (11), for $1 \le i \le n$.

Step 2. Calculate q_i by using (10).

Step 3. Check whether q_i's are relatively prime. If they are not, go to Step 1. Otherwise, we have found at least one possible solution in the form of $((q_1, k_1), (q_2, k_2), \cdots, (q_n, k_n))$.

Step 4. Randomly generate a message $M = (m_1, m_2, \cdots, m_n)$. Encrypt M by the Step 4 in Algorithm 3.2 into an integer C.

Step 5. Decrypt C into M'' by Step 1 in Algorithm 3.3 using the n pairs $((q_1, k_1), (q_2, k_2), \cdots, (q_n, k_n))$ obtained.

Step 6. If M'' and the M generated in Step 4 are equal, stop; otherwise go to Step 1 again.

Up to now, there seems to be no easy way of executing Step 1 (solving a Diophantine equation with degree 4). Even if we succeed, there is no guarantee that the q_i's found by us are relatively prime to one another. Therefore, it seems difficult to break our system in this way.

C. Attack Due to the Greatest Common Divisor of $\mathbf{s_i}$'s

Another ciphertext attack is to observe the greatest common divisor of s_i's. On intercepting the ciphertext C and the publicly known $s_1, s_2, \cdots, s_n$, the cryptanalyst hopes to decrypt C into M as in the Step 1 of Algorithm 3.3. Since the cryptanalyst has no legitimate (q_i, k_i)'s, m_i may be obtained by the following exhaustive searching steps.

Step 1. Compute t_i, for $i = 1, 2, \ldots, n$, as follows

$$t_i = \frac{\gcd(s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)}{\gcd(s_1, s_2, \cdots, s_{i-1}, s_i, s_{i+1}, \cdots, s_n)}$$

where gcd denotes the greatest common divisor.

Step 2. Compute $r_{ij} = j \cdot s_i \bmod t_i$, for $j = 1, 2, \cdots, w$, and $i = 1, 2, \cdots, n$, where $w = 2^b - 1$ if each submessage is of length b bits.

Step 3. Compute $h_i = C \bmod t_i$, for $i = 1, 2, \cdots, n$.

Step 4. Search h_i for $i = 1, 2, \cdots, n$, from the set $\{r_{1i}, r_{2i}, \cdots, r_{wi}\}$. If $h_i = r_{ki}$, then $m_i = k$.

From the above procedure, m_i seems to be deducible from C and $(s_1, s_2, \cdots, s_n)$. However, if we decompose the message into submessages of length 100 bits each; i.e., $b = 100$, then $w = 2^{100} - 1$. This number has magnitude of value about 10^{30}. If we use a computer that can test 10^6 numbers per second. It requires about 2.7×10^{16} years to complete the search for each h_i. The Step 4 of exhaustive searching in the above algorithm will be extremely impossible.

V. Conclusion and Discussion

A new public-key cryptosystem is investigated in this paper. The motivation of this attempt is trying to use real numbers for its dense property. However, if real numbers are used as keys, several disturbing problems, such as representation and precision will be encountered. With the help of integer functions, the possibility of using an integer as a key is increased significantly. That is, for a cryptanalyst who tries to break the cipher, he has to conduct an exhaustive search on a long list of integer numbers.

Further, we would make some discussion on the parameters used in the presented cipher scheme. By using a concept similar to that of block cipher [5], a sending message of length nb bits will be broken into n pieces of submessages with each b bits long. The time complexity needed to compute q_i's will be proportional to n^2 as n increases [2]. When q_i's are determined, k_i's can be chosen from 2) and 3) in the DK-conditions. Thus the time required to choose k_i's is proportional to n. Further, the time needed to find b_i's grows at the rate of $n(\log n)$ when q_i's and k_i's are determined.

From Section IV, we know that the execution time required, for a cryptanalyst to solve the corresponding problems, increases when n increases. Theoretically, the security of the presented scheme will be increased as n is large. For inatance, when $n = 100$ and $b = 100$, it will be rather difficult to solve the problems presented in Section IV. Further, let us estimate how large the C value is. We consider that the number of bits needed to store the product of the first n prime numbers is proportional to $n(\log n)$. Then the number of bits required to represent s_i is proportional to $n(\log n)$. In other words, the number of bits to represent a C value is proportional to $b + n(\log n) + (\log n)$, where b is the number of bits in each submessage. Since a sending message is of length bn bits. We conclude that the ciphertext expansion rate of the presented scheme is $O(\log n)$.

Finally, we would like to point out that the advantage of the presented scheme is that the encryption and decryption steps