PUBLIC-KEY CRYPTOSYSTEM BASED ON THE DIOPHANTINE EQUATIONS

Ioan Mang, Erica Mang
Department of Computers,
University of Oradea, Faculty of Electrical Engineering and Information Technology,
1, Universitatii St., 410087 Oradea, Romania, E-mail: emang@keysys.ro

***Abstract* – This study analyses the mathematical aspects of diophantic equations and the potential of using them in cipher public-key systems. There are also presented the algorithms written in C language that were used for implementing such a system**

***Keywords:* public-key algorithms, Diophantine equation, cryptosystem, encryption.**

I. INTRODUCTION

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.

In general, a Diophantine equation is defined as follows: We are given a polynomial equation $f(x_1,x_2,...,x_n)$ - 0 with integer coefficients and we are asked to find rational or integral solutions [6].

Throughout this paper, we shall assume that the solutions are nonnegative. For instance, consider the following equation:

$$3x_1 + 4x_2 + 7x_3 + 5x_4 = 78. \qquad (1)$$

The above equation is a Diophantine equation if we have to find a nonnegative solution for this equation. In fact, our solution is

$$(x_1, x_2, x_3, x_4)=(2, 5, 1, 9). \qquad (2)$$

A famous Diophantine equation problem is Hilbert's tenth problem, which is defined as follows: Given a system of polynomials $Pi(x_1, x_2, ..., x_n)$, $1 \le i \le m$, with integer coefficients, deter-mine whether it has a nonnegative integer solution or not. In and, it was shown that the Hilbert problem is undecidable for polynomials with degree 4.

It was shown in that the Hilbert problem is undecidable for polynomials with 13 variables [1]. Gurari and Ibarra also proved that several Diophantine equations are in NP-complete class.

II. THE UNDERLYING MATHEMATICS

Let w be some positive integer and the domain 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, 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 D.

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

1) q_i's are pairwise relative primes; i.e.

1. $(q_i,q_j) = 1$ for i j.
2. 2) $k_1 > w$ for i = 1,2, ..., n.
3. 3) $q_i > k_i w(q_i \bmod k_i)$, and
   $q_i \bmod k_i$ 0, for i = 1, 2, ..., 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. 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 = j.

Since q_u'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

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

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. Note that the average number of divisions performed by the extended Euclid's algorithm for finding b_i is approximately 0.843. ln (q_i) + 1.47. Secondly, compute

$$N_i = \lceil q_i/(k_iR_i) \rceil \qquad \text{for } i = 1, 2, ..., n.$$

Finally, compute

$$s_i = P_iN_i \bmod Q \;\; \text{where} \;\; Q = \prod_{i=1}^{n} q_i \qquad (3)$$

That is, we have a vector S = (s_1, s_2, ..., 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, ..., m_n) and S = (s_1, s_2, ..., s_n); i.e.,

$$C = E(S,M) = M*S = \sum_{i=1}^{n} m_is_i \qquad (4)$$

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

$$m_i = D((q_i,k_i),C) = \lfloor k_iC/q_i \rfloor \quad \text{for } i = 1, 2, ..., n \qquad (5)$$

Theorem 2.1 shows that (5) is the inverse function of (4). 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

$$ac < ax/(b + a). \qquad (6)$$

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

$$(ax/b + a) \leq x. \qquad (7)$$

Combining (6) and (7), we have that

$$a\lceil x/b \rceil < x \quad \text{if } x \geq ab / (b-a). \qquad (8)$$

Lemma 2.2:
Let $R_i = q_i \bmod k_i$. Then

$$k_iR_im_i \lceil q_i/(k_iR_i) \rceil \bmod k_iq_i = k_iR_im_i \lceil q_i/(k_iR_i) \rceil \qquad (9)$$

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 $q_i > k_i\, R_i 2\, m_i / (Ri\, (k_i - m_i))$.
That is, $x \geq ab / (b-a)$ is satisfied. By applying Lemma 2.1, it can be seen that $R_im_i\lceil q_i/(k_iR_i) \rceil < qi$. Therefore,

$$k_i\, R_i\, m_i \lceil q_i/(k_iR_i) \rceil \bmod k_i\, q_i = k_i\, R_i\, m_i \lceil q_i/(k_iR_i) \rceil \quad (10)$$

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

$$\lfloor k_iR_im_i\lceil q_i/(k_iR_i) \rceil / q_i \rfloor = m_i. \qquad (11)$$

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

$$\delta < \lfloor k_iR_im_i(q_i/(k_iR_i) + 1)/q_i \rfloor = mi \qquad (12)$$

and

$$\delta \geq \lfloor k_iR_im_i(q_i/(k_iR_i))/q_i \rfloor = mi. \qquad (13)$$

Furthermore, the right-hand side of (13) is identical to m_i and that of (12) is $\lfloor m_i + k_iR_im_i/q_i \rfloor$. On the other hand, since m_i is an integer and $k_iR_im_i / q_i < 1$, the right-hand side in (12) becomes

$$\lceil m_i + k_iR_im_i/q_i \rceil = m_i. \qquad (14)$$

Combining these two inequalities. we obtain that $m_i \leq \delta \leq m_i$. Finally, we have $\delta = m_i$, since δ is an integer.
Theorem 2.1: Let (q_1, k_1), (q_2, k_2), ..., 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 $s_i = P_iN_i'$ we have a vector

$$s = (s_1,s_2,...,s_n);\ \text{i.e.,}\ s_i = s_i \bmod Q, \ \text{for } i = 1,2,...,n.$$

Let

$$C' = M*S = \sum_{i=1}^{m} m_is_i = \sum_{i=1}^{m} m_iP_iN_i\ . \qquad (15)$$

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≠j j,

$$k_iC' \bmod k_iq_i = (k_i \sum_{i=1}^{n} m_iP_iN_i) \bmod k_iq_i = $$
$$= k_im_iR_i \lceil q_i/(k_iR_i) \rceil \bmod k_iq_i. \qquad (16)$$

Furthermore, by Lemma 2.2,

$$k_im_iR_i \lceil q_i/(k_iR_i) \rceil \bmod k_iq_i = k_im_iR_i \lceil q_i/(k_iR_i) \rceil \qquad (17)$$

That is, $k_iC' \bmod k_iq_i = k_im_iR_i \lceil q_i/(k_iR_i) \rceil$ for i=1,2,...,n. In other words,

$$k_iC' = y_ik_iq_i + k_im_iR_i\lceil q_i/(k_iR_i) \rceil. \qquad (18)$$

for some integers y_i.
Moreover,

$$k_iC'/q_i=y_ik_i+k_im_iR_i\left\lceil q_i/(k_iR_i)\right\rceil/qi. \qquad (19)$$

Hence

$$\left\lfloor k_iC'/q_i\right\rfloor=\left\lfloor y_ik_i+k_im_iR_i\left\lceil q_i/(k_iR_i)\right\rceil/q_i\right\rfloor= \\ =y_ik_i+\left\lfloor k_im_iR_i\left\lceil q_i/(k_iR_i)\right\rceil/q_i\right\rfloor \qquad (20)$$

By applying Lemma 2.3, we have

$$\left\lfloor k_iC'/q_i\right\rfloor = y_ik_i + m_i. \qquad (21)$$

Thus

$$m_i = \quad \bmod k_i. \qquad (22)$$

Second, let

$$Q=\prod_{i=1}^{n}q_i\ . \qquad (23)$$

then

$$C' \bmod Q = (\textstyle\sum_{i=1}^{n}m_is_i) \bmod Q = ((m_1s_1 \bmod Q) + \ldots \\ + (m_ns_n \bmod Q)) \bmod Q = (m_1\,(s_1 \bmod Q) + \ldots \\ + m_n(s_n \bmod Q))\bmod Q = (\sum_{i=1}^{n}m_is_i) \bmod Q = C \bmod Q.$$

That is, C’ = C(mod Q).
Let C’ = C + zQ, for some positive integer z.
We have

$$\left\lfloor k_iC/q_i\right\rfloor \bmod k_i =(\left\lfloor k_i(C'-zQ)/q_i\right\rfloor \bmod ki = \\ = (\left\lfloor k_iC'/q_i-k_izQ_i\right\rfloor) \bmod k_i = \bmod k_i \qquad (24)$$

In other words, $m_i=\left\lfloor k_iC'/q_i\right\rfloor \bmod k_i$.

III. THE CONSTRUCTION OF THE CRYPTOSYSTEM

In this section, the algorithms for constructing the cryptosystem, encrypting messages, respectively, are presented.
First, each user picks n pairs of parameters (q_1,k_1), (q_2,k_2),..., and (q_n,k_n) such that the DK-conditions are satisfied. Afterward,

$$Q=\prod_{j\neq i}q_j \qquad (25)$$

and

$$N_i=\left\lceil q_i/(k_i(q_i \bmod k_i))\right\rceil \qquad (26)$$

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, . . . n.
Let $P_i = Q_ib_i$ and $s_i = P_iN_i \bmod Q$, for i = 1,2,...,n, where

$$Q=\prod_{i=1}^{n}q_i \qquad (27)$$

Therefore, a vector

$$S=(s_1,s_2,\ldots,s_n)$$

is obtained. There the n-tuple S of intgers 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), ..., (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

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

where m_i is a b-bits submessage represented by a decimal number in the range of [0,2b-1].
Then $(m_1,m_2,\ldots,m_n)$ is enciphered by (4) 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 $(m_1,m_2,\ldots,m_n)$ by applying (5).

IV. 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.

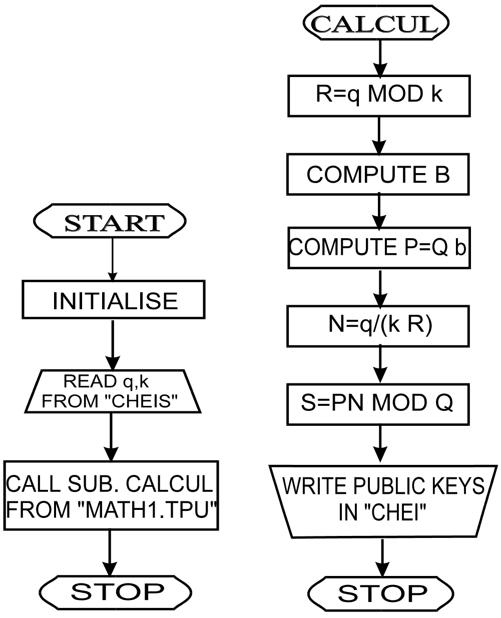

Fig. 1. Key Generating for Each User

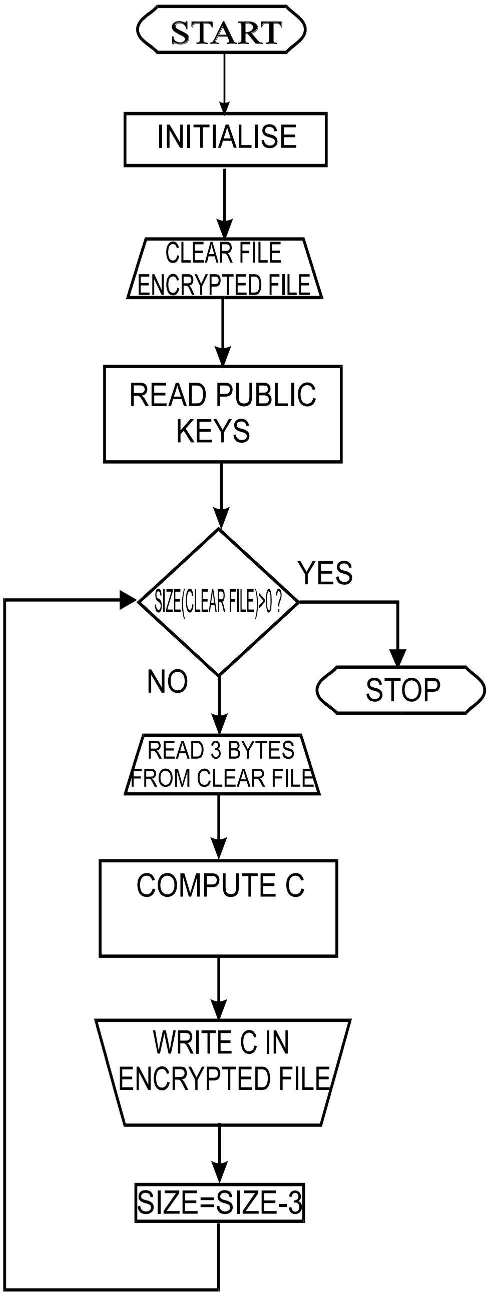

Fig. 2. - Encryption Procedure for Sender A

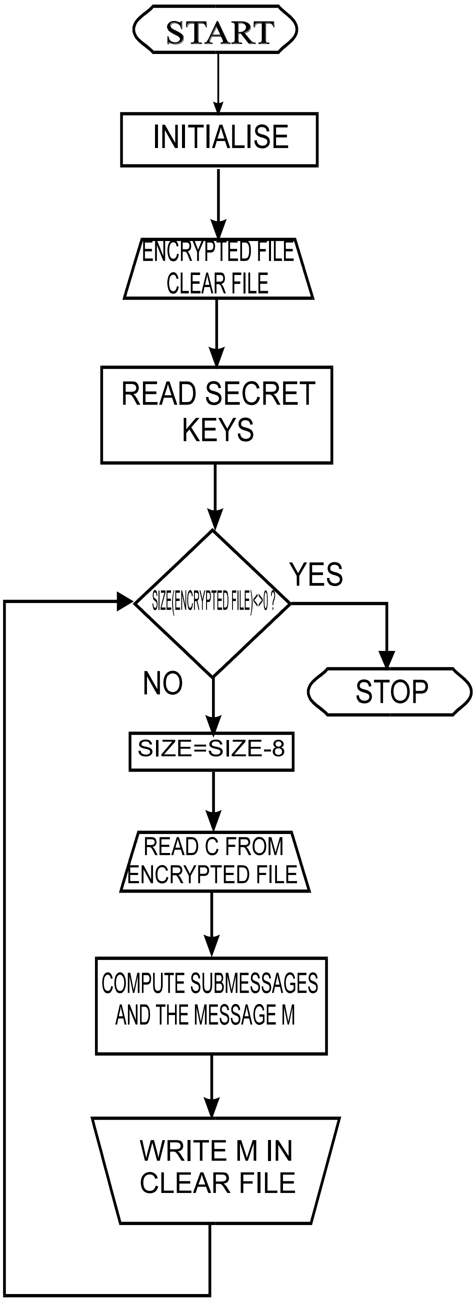

Fig. 3. - Decryption Procedure for Receiver B

REFERENCES

[1] D.E. Knuth: The Art of Computer Programming. Vol. 1: Fundamental Algorithms, second ed. Reading, MA: Addison-Wesley, (1980).

[2] D.E. Knuth: The Art of Computer Programming. Vol. 2: Seminumerical Algorithms, 2nd ed. Reading, MA: Addison-Wesley, (1981).

[3] S.P. Tung: Computational complexities of diophnatine equations with parameters, J. Algorithms, vol. 8, pp. 324-336, (1987).

[4] H.C. Williams: A modification of the RSA public-key encryption procedure, IEEE Trans. Information Theory, vol. 26, pp. 726-729 (1980).

[5] L.J. Hoffman: Modern Methods for Computer Security and Privacy, second edition, Printice-Hall, (1987)

[6] Waclaw Sierpinski, *"Elementary Theory of numbers",* Warszawa (1964)