Hybrid Public- Key Cryptosystem

Dr.Eng saad M. Khaleefah M.s.c Angham Kh. Hussein

Al-Turath College University

Abstract:

In this system a combination are made between two cryptosystems public-key . The first stage is the (RSA) System and the second is knapsack system. In this paper the analysis and evaluation of hybrid system are presented, and the computer program in C++ language of this system are applied.

Keyword : cryptosystem , knapsack, RSA, secure, public key.

بناء وتطبيق نظام تشفير باستخدام مفتاح تشفير عام متسلسل

د.سعد محمد خليفة م.م. انغام خالد حسين

كلية التراث الجامعة

الخلاصة:

في هذا النظام هناك دمج بين طريقتين من طرق التشفير باستخدام المفتاح العام.المرحلة الاولى هي (RSA) والمرحلة الثانية هي نظام (Knapsack). تم تقديم نتائج تحليلية لهذا النظام بالاضافة الى تطبيق للنظام بلغة (C^{++}) .

كلمات مفتاحية:

نظام تشفير , Knapsack, RSA, آمن, مفتاح تشفير عام.

1-Introduction

In this system a combination are made between two cryptosystems public key. The first stage is the RSA system and the second stage is knapsack public key. First the message is enter to the RSA public key cryptosystem and then the cipher is enter to knapsack system to encrypt the cipher again by this system. The final cipher is transmitted to the receiver via a secure channel, in the receiver the cipher is enter to knapsack system and then enter to RSA system, finally the original message is gained. In this paper the analysis and example of hybrid system is applied and a computer example is implemented in C++ language to represent the system [1].

2-Public-key RSA

In a ''public key cryptosystem'' each user places in a public file an encryption procedure E . That is, the public file is a directory giving the encryption procedure of each user. The user keeps secret the details of his corresponding decryption procedure D. These procedure have the following four properties [2]:

(a) Deciphering the enciphered form of a message M yields M. Formally, $D(E(M) = M$ ...[1]

(b) Both E and D are easy to compute.

(c) By publicly revealing E the user dose not reveal an easy way to compute D.
This means that in practice only he can decrypt messages encrypted with E, or compute D efficiently.

(d) If a message M is first deciphered and then enciphered, M is the result. Formally,

$$E(D(M) = M \ldots [2]$$

An encryption (or decryption) procedure typically consists of a general method and an encryption key. The general method, under control of the key, enciphers a message M to obtain the enciphered form of the message, called the ciphertext C. Everyone can use the same general method; the security of a given procedure will rest on the security of key. Revealing an encryption algorithm then means revealing the key [3].

3-Knapsack Public –Key Cryptosystem

This system is based on the knapsack problem. Given a knapsack of length C and a set of n-rods all of the same diameter as the knapsack but of lengths a_1, a_2 , a_3 , ..., a_n find a subset of the rods that completely fills the knapsack[3].

In general the only know solution is the enumeration technique. This technique is computationally infeasible for large n (e.g. n=100) the computing time could be several years. The knapsack system operates as follows[4]:

1- The user generates an n-integer $\acute{a}_1 , \acute{a}_2 , \ldots\ldots , \acute{a}_n$(easy solved knapsack problem) with the property that each element is greater than the sum of the preceding elements these integers is the deciphering key D. These integers are kept secret. Also he generates two large number m and w, such that w is invertible modulo m (i.e. gcd (m, w)=1), and $m>\sum \acute{a}_i$. These two integers also kept secret by the receiver.

2- Then he computes the integers $a_1 , a_2 , \ldots , a_n$ (the hard solved knapsack problem) via the relation:

$$a_i = \acute{a}_i * w \ mod \ m \ldots [3]$$

These integers are transmitted to the sender or stored in a public file. These integers is the enciphering key E which is public.

3- The sender converts the message into its binary representation, and divides this binary representation into blocks each of length n-bits. The encryption of the message is accomplished by encrypting each block.

Let $X_1, X_2, \ldots, X_n$ be one of these blocks, the encryption of this block is accomplished as follows:

$C = a_1 X_1 + a_2 X_2 + \ldots\ldots + a_n X_n \ldots [4]$

Which is the information transmitted via insecure channel to the receiver.

4- The receiver computes first w^{-1} via the relation:

$w\, w^{-1} = 1 \bmod m \quad \ldots [5]$

Then C' is computed by the following relation:

$$C' = C * w^{-1} mod\, m \quad \ldots [6]$$

Then the receiver begin to recover the X_i's by comparing C' with a_n'. If $C' > a_n'$ then set X_n equal to 1, otherwise X_n equal to zero. If $X_n = 1$ then the subtracts a_n from C a new value is found, then comparing this value with a_{n-1}, if the new value of C' is greater than a_{n-1}' then X_{n-1} is set equal to 1, otherwise X_{n-1} is set to zero. This process is repeated until the X_i's is computed. The knapsack problem $a_1', a_2', \ldots\ldots, a_n'$ with constant C is called the trap-door.

knapsack problem because it is hard to solve, however it has the same solution as the easy solved knapsack problem $a_1', a_2', \ldots\ldots, a_n'$ with constant C' [4].

4- Hybrid Cryptosystem

Since no techniques exist to prove that an encryption scheme is secure, the only test available is to see whether the enemy can think of a way to break it. It

is well known that the security of the RSA cryptosystem is based on the problem of factoring a large integer n into it's factors is computationally unfeasible[3]. A large number of factoring algorithms exist. Factoring large number algorithm is due to Richard Schroeppel (un published) who proposed the approximate factoring of prime number n of length t digits as:

$$Z = \ln(n)^{[\ln(n)/\ln(\ln(n))]^{1/2}} \ldots [7]$$

Where Z is the number of multiplications.

In the knapsack public key

$(X_1, X_2, X_3, \ldots, X_n)$ such that

$$Y = \sum_{j=1}^{N} a_j . X_j \quad \ldots [8]$$

Where Y is the information transmitted over an insecure channel so that any eavesdropper has access to it. Since the vector $(a_1, a_2, \ldots, a_N)$ is made public, the eavesdropper has access to it. The only solution available is an enumeration-method search over all 2^N possible vectors of X.Break the trapdoor knapsack cryptosystem is[1]:

$$Z = N \cdot 2^N \ldots [9]$$

For a highly secure cryptosystem, it is necessary to use a long n in the RSA cryptosystem-e.g. n consists of 100 digits. The idea of the hybrid system is to use two stages of encryption, the first stage is a RSA cryptosystem and the second stage is the cryptosystem. This arrangement is shown in the Figure (1). The enciphered text at the output of the first system is considered as a message for the second system. In the receiver, the enciphered transmitted information is decrypted using the algorithm firstly and the knapsack algorithm secondly (The decryption order is opposite to the encryption order.)

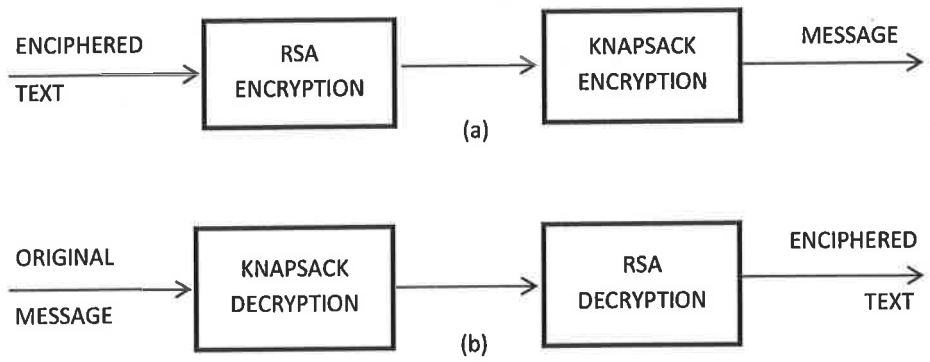

Fig (1) Hybrid cryptosystem: (a) transmitter; (b) receiver

The security of this system is evaluated by the number of arithmetic operations required to break it Since, in the knapsack cryptosystem, the knapsack vector has 2^N possibilities, then for each possibility the enemy is forced to decrypt the RSA algorithm[5]. The required number of arithmetic multiplications for breaking the hybrid system, using a prime number n of length t digits in the RSA cryptosystem, and a knapsack vector of length N.

$$Z = N.2^N . \ln (n)^{[\ln(n)/\ln(\ln(n))]1/2} \quad \ldots[10]$$

Where Z is no. of multiplicative operations, n is the no. of RSA system, and N is the length of vector of knapsack.

4.1 Hybrid Public-Key Cryptosystem

Message is encrypted in two-stage and the order of encryption is:

$$Z = N.2^N . \ln(n)^{[\ln(n)/\ln(\ln(n))]^{1/2}} \ldots[11]$$

Table 2 evaluates the complexity of the RSA cryptosystem using a prime number of length n, a knapsack cryptosystem having a vector length N, and a hybrid cryptosystem[6]. The last column in Table (1) represent the required number of multiplicative operations of hybrid system.

Table (1): Required number of multiplications

t	N	Knapsack	RSA	Hybrid	Required length using RSA
10	10	1.02×10^{4}	4.9×10^{3}	4.99×10^{7}	31
20	20	2.09×10^{7}	5.85×10^{5}	1.22×10^{13}	76
30	30	3.22×10^{9}	2.68×10^{7}	8.62×10^{16}	117
40	40	4.39×10^{13}	7.31×10^{8}	3.20×10^{22}	192
50	50	5.62×10^{16}	1.41×10^{10}	7.92×10^{26}	260
100	100	1.26×10^{32}	2.34×10^{15}	2.94×10^{47}	702
200	200	3.21×10^{62}	1.2×10^{23}	3.85×10^{85}	1999

5-Results:

A computer example are applied in C++ programming language ,a message " cipher text" is used as input to the hybrid cryptosystem and the encryption and decryption results are shown in the table (2):

Table (2) :Encryption and Decryption Results

Input	c	i	p	h	e	r	t	e	x	t
Encryption	169	169	428	169	169	3	478	3	169	169
Decryption	64	64	162	64	64	1	181	1	64	64
Output	c	i	p	h	e	r	t	e	x	t

Where the encryption represent the encryption of each letter by using hybrid system i.e by using RSA and than Knapsack encryption methods ,and the result

of decryption are result from using RSA and then Knapsack and the result of knapsack is converted from binary to decimal which give the equivalent letter.

6-Conclusions:

A new cryptosystem which is mixed the RSA with knapsack have been developed the basic benefit of this hybrid system is have a higher security and higher speed .while have the following disadvantages:

1-Complexcity of the system increased.

2-Big memory size are required.

3-Encryption and decryption time increased.

The tradeoff between the length of (n) of RSA and the time required to obtain the required security.

REFRENCES:

[1] Saad M. KH., Jafar Wadi (New Public Key) int. j. system.sci . 1990 vol. 21.

[2] Chinstof parry (Introduction To Public Key), spring 2009

[3] Special Lecture From Internet About Public key Cryptography ,2008

[4] J. Katz Y. Lindel (Introduction To Modern Cryptography) CRC Pres Is BN 1-5-8488-551-3, 2007

[5] B. Schnerier (Applied Cryptography) 2n□ Edition Jon Wiley and sons , newyork 1996 .

[6] Knuth D.E (The art of Computer Programming) vol. 2 1968.