Implementation of Hybrid Combination of knapsack-RSA Cryptography Algorithm Using Mat lab Code

Dr. Saad. M. Khalipha   Dr. Mohamed N. Majeed   Ali A. Abdul-Hameed

Abstract:
A serial combination of knapsack and RSA (Rivest–Sharmir-Adelman) cryptosystem can be represented as a secure system. The results that achieved from using this technique is the encrypted message which acquires a higher security and accuracy when compared with knapsack and RSA crypto systems individually in modern communication systems to protect the information transmitted over an insecure communication channel.

Here software using Mat lab code algorithm is built for implementing the hybrid combination of Knapsack-RSA algorithm, with successful results.

Key words: Cryptosystem, Knapsack, RSA, Security.

Knapsack-RSA تنفيذ خوارزمية التشفير الهجينة

باستخدام برنامج ماتلاب

د. سعد محمود خليفة     د. محمد نوري مجيد     م. م. علي أحمد عبد الحميد
كلية المأمون الجامعة – قسم هندسة الاتصالات

المستخلص:
يتم الربط المتوالي لتقنيات التشفير المستخدمة في منظمات الاتصالات بحيث تكون البيانات المرسلة والبيانات مشفرة باستخدام خوارزمية Knapsack الناتجة يتم تشفيرها مرة أخرى باستخدام خوارزمية RSA كنظام أمان.
والنتيجة تصبح البيانات المرسلة أكثر أمانا من الاستراق، واكثر دقة مقارنة مع تلك المنظمات التي يستخدم بها خوارزمية Knapsack أو RSA فقط.
في هذا نقدم خوارزمية باستخدام برنامج الماتلاب لبناء وتصميم الخوارزمية الهجينة من البحث Knapsack – RSA
1. Introduction:

In any communication system including internet, satellite and mobile, it is impossible to prevent the important or sensitive information from eavesdropping or losses when the information is broadcasted through the channel (wire or wireless). So security of information has become increasingly important for any application.

Cryptography is the science that stands for methods used to transform data and hide its contents far from wrong hands. Cryptography defines a pair of two techniques that which grantees data transformation called encryption and decryption.

Encryption is the process of transformation data (plaintext) to be transmitted using algorithm (called cipher), to unreadable form (cipher text). while decryption is the process to convert the encrypted data to the original form.

The message is encrypted with public key and can only be decrypted by using the private key. So the encrypted message cannot be decrypted by any one who knows the public key and thus secure data broadcasting (communication) is possible.

The primary goal of this paper is to implement the hybrid combination of RSA-Knapsack algorithm using Mat lab codes.

2. RSA Cipher

An RSA public key and a corresponding private key can be generated according to the following procedure.

At first generate two large prime numbers p and q, each the same size, next compute n=pq and \( \varphi = (p-1)(q-1) \), then select a random integer e, less then \( \varphi \) which is relatively prime to it, such that \( \gcd(e, \varphi )=1 \). Finally use the extended Euclidean algorithm to compute the unique integer d, where 1 < d < \( \varphi \), such that \( ed = 1 \mod \varphi \). Thus the public key is the pair (e, n), and the private key is d [1][2].

A restriction on selecting p and q is that they should be about the same bit length, and sufficiently large. For example, if a 1024-bit modulus n is to be used, then each of p and q should be about 512 bit in length, and the difference p-q should not be too small. Finally, p and q must be strong primes. A prime p is strong prime if the following conditions are satisfied [3].

a) P -1 has a large prime factor, called r
b) \( P + 1 \) has a large prime factor.
c) \( R - 1 \) has a large prime factor.

3. Knapsack Cipher

A knapsack is a vector of length \( N(a_1, a_2, a_3, \ldots, a_N) \) that may be used for encryption. The sender converts their message into a string of binary numbers, which is broken into blocks of \( N \)-bits, and each block \( X = (X_1, X_2, X_3, \ldots, X_n) \), i.e. enciphered by performing the dot product with the knapsack vector i.e.

\[
Y = a_1X_1 + a_2X_2 + a_3X_3 + \ldots + a_NX_N
\]

The Merkle-Hellman trapdoor-knapsack algorithm for encryption involves the following procedure.

a) User \( i \) randomly chooses two large integers \( w \) and \( m \) (\( m > w \)), such that \( \gcd(w, m) = 1 \).
b) In verifying that \( \gcd(w, m) = 1 \), user \( i \) finds \( c \) and \( r \) such that

\[
1 = cw + rm
\]

\[ R_m(cw) = 1 \]
c) User \( i \) then randomly chooses an easy knapsack \((a'_1, a'_2, a'_3, \ldots, a'_N)\) with the property of superincreasing that is

\[
a'_i > \sum_{j=1}^{i-1} a'_j
\]

And

\[
m > \sum_{j=1}^{N} a'_j
\]

The elements of the hard knapsack are computed from

\[
a_j = R_m(wa'_j)
\]

d) The hard knapsack \((a_1, a_2, a_3, \ldots, a_N)\) is listed in the public directory
e) Another user sends \( X = (X_1, X_2, X_3, \ldots, X_N) \) to \( i \) as

\[
Y = \sum_{j=1}^{N} a_j \cdot x_j
\]
f) User \( i \) decrypts \( Y \) by solving the easy knapsack problem for \( Y' = R_m(CY) \). That is \( Y' \) is calculated from \( Y \) and then decoded using the secret easy knapsack.
g) User \( i \) begins to recover the transmitted block of the message \((X_1, X_2, X_3, \ldots, X_N)\) by comparing \( Y' \) with \( a'_N \), if \( Y' > a'_N \), and then \( X_N = 1 \), or otherwise \( X_N = 0 \). If \( X_N = 1 \) then \( a'_N \) is
subtracted from $Y'$ to compute a new value of $Y'$. Again $Y'$ is compared with $\alpha_{n-1}'$. This process is repeated until $X_1$ is deciphered[4]

### 4. The Proposed System:

It is well known that the most important topic in the field of communication systems including internet, satellite or mobile is a security of exchanging the sensitive information. The previous two ciphering techniques are maintained, RSA and knapsack cryptosystems. The security of RSA cryptosystem is based on the problem of factoring a large number. A factoring of $n$ would enable the eavesdropper to break the algorithm. The factors of $n$ enables the eavesdropper to compute $\varphi(n)$, and thus $d$. The security of the knapsack cryptosystem is based on finding a binary vector $(X, X_2, X_3 ....X_N)$ such that

$$Y = \sum_{j=1}^{N} \alpha_j \cdot X_j$$

Where $Y$ is the information transmitted over an insecure channel so that any eavesdropper has access to it. The only solution available is an enumeration method search over all $2^N$ possible vectors of $X$. Since each possibility requires ($N$-number of multiplication) operations, the total number of arithmetic multiplication required to break the trapdoor knapsack cryptosystem is $Z + N > 2^N$

For a highly secure cryptosystem, it is necessary to use a long $n$ in the RSA cryptosystem, or a long knapsack vector $(X, X_2, X_3 ....X_N)$ in the knapsack cryptosystem. However, increasing the security yields the following disadvantages:

1. Complexity of the system increases.
2. Requirement on the memory size increases.
3. Encryption and decryption time increases.
5. Requirement of searching for a long prime number.

To overcome this problem, a hybrid combination is suggested as shown in Fig(1), which offers extremely good security with less complexity and less time required for encryption-decryption process in compression with RSA and knapsack individually. The idea behind of this system is to use two stages of encryption. The first stage is a
knapsack cryptosystem and the second stage is the RSA cryptosystem as shown in the figure. The enciphered text at the output of the first stage is as input message for the second stage. In the receiver, the enciphered transmitted message is decrypted using the RSA algorithm firstly and the knapsack algorithm secondly. Thus the decryption order is opposite to the encryption order. [5][6]

![Diagram of Hybrid Combination Cryptosystem](image)

**Fig (1) Hybrid Combination Cryptosystem; (a) Transmitter (b) Receiver**

5. Software Implementation
A proposed algorithm [Appendix-A] is written using Matlab code with two stages. At the first stage knapsack algorithm is used to encrypt the transmitted message, and then the encrypted message is encrypted once again using the RSA algorithm. The encryption-decryption processes are shown in Fig (2) and Fig (3).
Fig (2) The Encryption Process

Knapsack Ciphering Stage

- Read Data (Text, Image)
- Input Knapsack Sequence
  \[ a_i > \sum_{j=1}^{i-1} a_j \text{ for each } 2 \leq i \leq n \]
- Convert the message \( m \) as a binary String of length \( n \)
  \[ m = m_1, m_2, \ldots \ldots \ldots m_n \]
- Compute Cipher text \( C \)
  \[ C_k = \sum_{i=1}^{n} m_i \times a_i \]

Send Ciphered Message \( C \) to Next Ciphering Stage

RSA Ciphering Stage

- Choose Prime Numbers \( p \) and \( q \)
  \[ n = p \times q \]
  \[ \varphi = (p-1)\times(q-1) \]
- Choose \( e \), Where \( 1 < e < \varphi \)
- Compute \( d \) and \( d = e^{-1} \mod \varphi \)
- The ciphered message is
  \[ c_R = c_k^e \mod n \]

Here the output message is ciphered
Fig (3) The Decryption Process

Ciphered Message

**RSA deciphering**

\[ C_R = C_R^d \mod n \]

First Stage Deciphering

**Knapsack deciphering**

Select a Random Integer \( W \)

Select a Random Integer \( M \)

\[ M > \sum_{i=1}^{p} a_i \]

\[ M > a_1 + a_2 + ... + a_n \]

Select a Random Permutation \( \pi \) of \( \{ 1, 2, 3, 4, ..., n \} \)

\( \pi(1) = 3, \pi(2) = 6, \pi(3) = 1, ... \)

Compute \( d \)

\[ d = W^{-1} \times C_R \mod M \]

\[ d = m_1 r_1 + m_2 r_2 + ... + m_n \]

Apply \( \pi^{-1} \) (Return the Message Bits)

Second Stage Deciphering (Return The original)

Fig (3) The Decryption Process
5. Conclusions

1. A hybrid combination Knapsack-RSA cryptosystem offers higher security.

2. In a hybrid combination Knapsack-RSA cryptosystem, the encryption-decryption process time delay is longer than that of RSA and Knapsack cryptosystem individually.

References: