A New Trapdoor Knapsack Public Key Cryptosystem.

R.M.F. Goodman, B.Sc., Ph.D., C.Eng., M.I.E.E.
and
A.J. McAuley, B.Sc.

Department of Electronic Engineering
University of Hull
HULL HU6 7RX
U.K.

20 July 1984

Keywords : cryptography, ciphers, codes, knapsack problem, public key.

Abstract

This paper presents a new trapdoor-knapsack public-key-cryptosystem. The encryption equation is based on the general modular knapsack equation, but unlike the Merkle-Hellman scheme the knapsack components do not have to have a superincreasing structure. The trapdoor is based on transformations between the modular and radix form of the knapsack components, via the Chinese Remainder Theorem. The resulting cryptosystem has high density and has a typical message block size of 2000 bits and a public key of 14K bits. The security is based on factoring a number composed of 256 bit prime factors. The major advantage of the scheme when compared with the RSA scheme is one of speed. Typically, knapsack schemes such as the one proposed here are capable of throughput speeds which are orders of magnitude faster than the RSA scheme.

T. Beth, N. Cot, and I. Ingemarsson (Eds.): Advances in Cryptology - EUROCRYPT '84, LNCS 209, pp. 150-158, 1985.

List of Principal Symbols

a_i = a published knapsack component.

a'_i = a secret knapsack component.

$\underline{a}$ = the public knapsack vector = $(a_1 , a_2 , \ldots , a_n)$.

$\underline{a}'$ = the secret knapsack vector = $(a'_1 , a'_2 , \ldots , a'_n)$, also transformable to the secret knapsack matrix.

$a_j^{(i)}$ = $a_j \bmod p_i$ = residue of the j th knapsack component modulo the i th prime.

D = density of the cryptosystem.

g = number of bits in $x_{i,max}$, the message sub-blocks.

h = number of bits in $p_{i,min}$.

K = the number of distinct secret matrices $\underline{a}'$.

n = the number of knapsack components, also, the number of primes p_i .

p_i = a prime number.

$\underline{p}$ = a set of n distinct primes = $(p_1 , p_2 , \ldots , p_n)$.

p = $\prod_{i=1}^{n} p_i$ = the product of n distinct primes .

PK = number of bits in the public key.

r = number of bits in $\left\{ \sum_{j=1}^{n} a_j'^{(i)} \right\}_{max}$.

S = the cryptogram = $\sum_{i=1}^{n} a_i \cdot x_i$.

S' = the transformed cryptogram = $S \cdot W^{-1} \bmod p$. also equal to $(S'^{(1)}, S'^{(2)}, \ldots , S'^{(n)})$ in modular form.

W = a secret modular multiplier, relatively prime to p .

$\underline{x}$ = the message vector = $(x_1 , x_2 , \ldots , x_n)$.

Introduction

Public-key-cryptosystems have received considerable attention over the last few years (Diffie and Hellman 1976, ref.1.). This is because such systems offer secure communications without the need for prior key distribution, and the possibility of digital signatures. The two most important schemes are the RSA scheme (Rivest, Shamir, and Adelman 1978, ref.2.), and the Trapdoor-Knapsack scheme (Merkle and Hellman 1978, ref.3.). Of these the Knapsack scheme has fallen into disfavour because of successful attacks on the original Merkle-Hellman scheme. Specifically, the attacks have not been on the encryption equation which appears secure, but on the fact that the knapsack components are transformations of a superincreasing sequence (Desmet 1982, ref.4). In addition, it has been shown that if the density of the knapsack is low, where density is loosly defined as the ratio of messagetext bits to cryptogram bits, then even non-superincreasing knapsacks are insecure (Brickell 1983, ref.5., Lagarias and Odlyzko 1983, ref.6.). Despite these problems knapsack schemes have one major practical advantage over the RSA scheme, and that is speed. This is because the encryption and decryption processes used are intrinsically faster than performing the modular exponentiations needed in the RSA. Typically, knapsack schemes can operate at throughput rates of 20Mbits/sec, whereas the RSA is limited to about 50Kbits/sec, using current technology.

The new trapdoor-knapsack presented in this paper uses the general modular knapsack equation (eqn. 1) , and does not require the knapsack components to be superincreasing. In addition, the system parameters can be chosen to give a very high density secure cryptosystem. The trapdoor is based on being able to transform between the radix and modular representations of the subset sums via the Chinese Remainder Theorem (Knuth 1968, ref.7.). The system bears a resemblance to the Lu - Lee (1979, ref.8.) system, but whereas their cryptosystem is linear and has been shown to be insecure (Goethals and Couvreur, 1980, ref.9.), ours is based on the general modular knapsack equation, which to date has not been generally broken.

The New Trapdoor

The general modular knapsack equation is given by

$$S = \sum_{i=1}^{n} a_i \cdot x_i \mod p \,. \qquad \text{eqn. 1.}$$

When used for cryptography, the a's are the n published knapsack components, p is a published modulus, and the x's are the message bits. In the binary knapsack the x's are 0 or 1, but in the general knapsack they are g bit numbers. The subset sum S is the cryptogram which is sent to the legitimate user, who is the only one who can unwind the cryptogram back to the original x's.

Let $(p_1 , p_2 , \ldots , p_n)$ be a set of prime integers whose product is given by

$$p = \prod_{i=1}^{n} p_i \quad , \text{ and where } \quad a_j^{(i)} = a_j \mod p_i$$

is the residue of the j th knapsack component modulo the i th prime.
Then by the Chinese Remainder Theorem

$$a_j \longleftrightarrow a_j^{(1)} , a_j^{(2)} , \ldots , a_j^{(n)}$$

is a bijective mapping. That is, the transformation is one-to-one for all a's between 1 and p-1. Thus if the factorisation of p is kept secret, then only the legitimate user will be able to transform the radix representation of the knapsack components into their modular representation. This forms the trapdoor. Let us now choose a set of n knapsack components and express them in both radix and modular form:

$$\underline{a}' = \begin{array}{l} a_1' \longleftrightarrow a_1'^{(1)}, a_1'^{(2)}, \ldots , a_1'^{(n)} \\ a_2' \longleftrightarrow a_2'^{(1)}, a_2'^{(2)}, \ldots , a_2'^{(n)} \\ \vdots \\ a_n' \longleftrightarrow a_n'^{(1)}, a_n'^{(2)}, \ldots , a_n'^{(n)} . \end{array} \qquad \text{eqn. 2.}$$

Let us then disguise the trapdoor by forming a new set of knapsack components via the modular multiplication

$$a_j = a_j' \cdot W \mod p \qquad \text{eqn. 3.}$$

where W and p are relatively prime, and W^{-1} is the multiplicative inverse of W , modulo p.

We now publish p , and the modified knapsack components ($\underline{a}$) in radix form. This is the public key. The factorisation of p and and the integer W are kept secret, and hence so is the modular representation of the a'.

Now let $$p_{i,min} \geq 2^h \qquad \text{eqn. 4.}$$

that is, the primes are at least h+1 bit numbers.

Let $$x_{i,max} < 2^g \qquad \text{eqn. 5.}$$

that is, the message blocks are g bit numbers.

And let $$\left\{ \sum_{i=1}^{n} a'^{(i)}_j \right\}_{max} < 2^r \qquad \text{eqn. 6.}$$

that is, the columns of $\underline{a}'$ sum to an r bit number.

In order to ensure that the encryption equation has a unique decryption, we must ensure that the message to ciphertext transformation $\underline{x} \longrightarrow S$ is injective. To guarantee this we must have

$$h \geq r + g \qquad \text{eqn. 7.}$$

which also ensures that modular multiplication is equivalent to matrix multiplication :

$$(S'^{(1)} , \ldots , S'^{(n)}) = (x_1, \ldots , x_n) \begin{Bmatrix} a'^{(1)}_1 , a'^{(2)}_1 , \ldots , a'^{(n)}_1 \\ \vdots \\ a'^{(1)}_n , a'^{(2)}_n , \ldots , a'^{(n)}_n \end{Bmatrix}$$

i.e. $$\underline{S}' = \underline{x} \cdot \underline{a}'$$

and that the transformation can be inverted (provided the matrix $\underline{a}'$ is non-singular) via

$$\underline{x} = \underline{S} \cdot \underline{a}'^{-1}. \qquad \text{eqn. 8.}$$

The cryptosystem then operates as follows. A user wishing to send us a message forms the ciphertext

$$S = (x_1 . a_1 + x_2 . a_2 + \ldots + x_n . a_n) \mod p$$

via equation 1. We compute S' via

$$S' = S \cdot W^{-1} \mod p$$

and express in modular form via our known factorisation of p :

$$S' \longleftrightarrow (S'^{(1)}, S'^{(2)}, \ldots , S'^{(n)})$$

we then apply $\underline{x} = \underline{S}' \cdot \underline{a}'^{-1}$ and hence recover the message.
The cryptanalyst must either break the factorisation of p or attack the trapdoor in some other manner.

A Small Example

We now give an example of the above method using n=3. The example is of course too small for security.

Let n = 3 and define $\underline{p}$ = (37, 41, 43) , hence p = 65231, and h = 5 (eqn. 4). Choose g = 2 , that is, the message components are two bit numbers. This dictates that r = 3 via equation 7. ($h = 5 \geq 3 + 2$). Choose n = 3 knapsack components which satisfy equation 6, that is, the columns of $\underline{a}'$ add to <8 , and express in both modular and radix form:

$$\underline{a}' = \begin{matrix} a'_1 = (3 , 1 , 1) & \longleftrightarrow & 125174 \\ a'_2 = (1 , 5 , 3) & \longleftrightarrow & 151664 \\ a'_3 = (2 , 1 , 2) & \longleftrightarrow & 122509 \ . \end{matrix}$$

Now choose W = 6553 which is relatively prime to p = 65231. Perform the modular multiplication of equation 3 , and publish the resulting knapsack components :

$$a_1 = 50628$$

$$a_2 = 59907$$

$$a_3 = 3560$$

and the modulus $p = 65231$.

Compute the inverse $W^{-1} = 2618$ via Euclid's algorithm and invert the matrix $\underline{a}'$:

$$\underline{a}'^{-1} = (1/16) \begin{Bmatrix} +7 , -1 , -2 \\ +4 , +4 , -8 \\ -9 , -1 , +14 \end{Bmatrix} .$$

To transmit the 6 bit message $\underline{x}$ = (1 , 2 , 3) a user computes the ciphertext

$$\begin{aligned} S &= (1 \ . \ 50628) + (2 \ . \ 59907) + (3 \ . \ 3560) \\ &= 181122 \\ &= 50660 \mod 65231 \ . \end{aligned}$$

Using the secret W^{-1} the receiver computes

$$\begin{aligned} S' &= 50660 \ . \ 2618 \mod 65231 \\ &= 13257 \mod 65231 \end{aligned}$$

and using the secret $\underline{p}$ is able to transform into modular form :

$$\underline{S}' = (11 , 14 , 13) \longleftrightarrow 13257 \ .$$

From equation 8, the receiver computes :

$$16 \ . \ \underline{x} = (11 , 14 , 13) \begin{Bmatrix} +7 , -1 , -2 \\ +4 , +4 , -8 \\ -9 , -1 , +14 \end{Bmatrix}$$

giving $\underline{x} = (1 , 2 , 3)$ as transmitted.

Practical Constraints

We now choose the values for n, r, g, and h needed to give a secure practical cryptosystem.

In order to present a large knapsack problem we set

$$n \cdot g \geq 256 \quad . \qquad \text{eqn. 9.}$$

The value of n is influenced by the fact that the general knapsack problem is not as secure as the binary knapsack because the least significant bits of the message are not as well hidden. We have reduced the problem by performing the reduction mod p , but we must still set a limit, say

$$n > 5 \quad . \qquad \text{eqn. 10.}$$

In order to protect the trapdoor and ensure that the published p is not factored we set

$$h \geq 255 \qquad \text{eqn. 11.}$$

so that the primes are at least 256 bit numbers.

To ensure sufficient randomness in the knapsack components we need to bound the number of valid matrices $\underline{a}'$, which we call K. If we assume that any number $1 \leq a_i' \leq 2^r$ can be chosen to be a knapsack component then the number of different column vectors that can be chosen is 2^{nr}, and thus

$$K = 2^{n^2 r} \quad .$$

However, because of the restriction on the sum of the column vectors imposed by equation 6, not all of these matrices are acceptable. Let us develop a conservative lower bound on K by employing an averaging argument. Assume that all knapsack components are chosen so that

$$1 \leq a_i' \leq \frac{2^r}{n} \quad .$$

This guarantees that all the resulting matrices will satisfy equation 6. The number of valid column vectors that can be chosen in this way is

$$2^{nr} \cdot \left\{ \frac{1}{n} \right\}^n \quad .$$

Which gives

$$K > 2^{n^2 r} \cdot \left\{ \frac{1}{n} \right\}^{n^2} \quad . \qquad \text{eqn. 12.}$$

To ensure sufficient randomness in the choice of knapsack components we require say $K \geq 2^{128}$. Taking logs of equation 12 we get :

$$n^2 \, (r - \log_2 n \,) \geq 128 \quad . \qquad \text{eqn. 13.}$$

The value of r is influenced by several factors. If r is small then the knapsack components will have a small remainder when divided by a factor of p (Goethals and Couvreur 1980, ref. 9.) This has been allowed for by the disguising modular multiplication (eqn. 3); but r must be large enough to ensure that no knapsack component has the same remainder modulo any prime factor. A loose lower bound falls out from equation 13. That is,

$$r > \log_2 n$$

but, if r is much less than n , then the choice of knapsack components is severely reduced by equation 6. Thus we set

$$r \geq n \ . \qquad \text{eqn. 14.}$$

The density of the cryptosystem is given by :

$$D = \frac{g}{(h + 1)}$$

if we assume the primes are all exactly h+1 bit numbers. Now, in order to minimise the redundancy of the scheme and to increase the resistance to low-density attacks, h should be as small as possible. Thus we set eqn. 7 to:

$$h = r + g$$

so that

$$D = \frac{g}{g + r + 1}$$

Thus to maximise D, we must keep r small. From equation 14 we should set r = n, and if we then set n = r = 7, we satisfy both equation 13 and equation 10.

The size of the public key is given by :

$$PK = n.(n + 1).(h + 1) \ ,$$

and in order to keep this small we must keep h small. So let us set eqn. 11 to

$$h = 255$$

which gives $g = 255 - 7 = 248$.

The size of the basic message block is then :

$$n \ . \ g = 1736 \text{ bits} \ ,$$

which certainly satisfies eqn. 9.

The final system parameters are then : n = r = 7, g = 248, h = 255 which gives D = 0.97 and PK = 14336 bits.

Conclusions

In this paper we have presented a new public key cryptosystem based on the general modular knapsack problem. Its security is not based on disguising a superincreasing sequence, but on the difficulty of factoring a number with seven 256 bit prime factors, and on a knapsack problem with a typical

density of 0.97 and a block size of 1736. The knapsack nature of the system ensures that fast encryption and decryption are possible when compared with the RSA public-key-cryptosystem. In addition, the size of the public key which is typically 14Kbits is not excessive. It may be possible to attack the trapdoor information more directly, but we can see no productive method of doing this. The only successful attacks on dense trapdoor-knapsacks to date have been on the security of the superincreasing sequence. Our method does not require this. However, it may turn out that all injective trapdoor knapsacks are solvable in polynomial time, in which case all such schemes are useless for cryptography.

References

1. W. Diffie and M. Hellman, "New directions in cryptography", IEEE Trans. on Information Theory, IT-22, pp 644-654, Nov. 1976.
2. R. Rivest, A. Shamir, and L. Adelman, "On digital signatures and public key cryptosystems", Comm. of the ACM, Vol. 21, No. 2, pp 120-126, Feb 1978.
3. R.C. Merkle and M.E. Hellman, "Hiding information and signatures in trapdoor knapsacks", IEEE Trans. on Information Theory, IT-24, pp 525-530, Sept. 1978.
4. Y. Desmet, J. Vandewalle, and R. Govaerts, "A critical analysis of the security of knapsack public key cryptosystems", IEEE Symp. on Information Theory, Les Arcs, France, June 1982.
5. E.F. Brickell, "Solving low density knapsacks", Sandia National Laboratories, Albuquerque, New Mexico, USA, 13p, 1983.
6. J.C. Lagarias, and A.M. Odlyzko, "Solving low-density subset sum problems", Bell Laboratories, Murray Hill, New Jersey, USA, 38p, 1983.
7. D.E. Knuth, The Art of Computer Programming, Vol. 1., "Fundamental Algorithms", Addison-Wesley, 1968.
8. S.C. Lu, and L.N. Lee, "A simple and effective public key cryptosystem", Comsat Tech. Rev., Vol. 9., pp15-24, Spring 1979.
9. J.M. Goethals, and C. Couvreur, "A cryptanalytic attack on the Lu-Lee public key cryptosystem, Phillips J. Res., Vol. 35, pp. 301-306, 1980.