New trapdoor-knapsack public-key cryptosystem

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

Indexing terms: *Digital communication systems, Cryptography*

Abstract: The 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 security is based on factoring a number composed of 256 bit prime factors. The resulting cryptosystem has high density, approximately 30% message expansion and a public key of 14 Kbits. This compares very favourably with the Merkle–Hellman scheme which has over 100% expansion and a public key of 80 Kbits. 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.

List of principal symbols

a_i = a published knapsack component
a'_i = a secret knapsack component
$\boldsymbol{a}$ = the public knapsack vector = $(a_1, a_2, \ldots, a_n)$
$\boldsymbol{a}'$ = 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 jth knapsack component modulo the ith prime
D = density of the cryptosystem
g = number of bits in $x_{i,\,max}$, the message subblocks
h = number of bits in $p_{i,\,min}$
K = the number of distinct secret matrices $\boldsymbol{a}'$
n = the number of knapsack components; also, the number of primes p_i
p_i = a prime number
$\boldsymbol{p}$ = 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 $\left(\sum_{j=1}^{n} a_j^{\prime(i)}\right)_{max}$
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
$\boldsymbol{x}$ = the message vector = $(x_1, x_2, \ldots, x_n)$
v = number of bits in the random-message tail
E = efficiency of the cryptosystem

1 Introduction

Public-key cryptosystems have received considerable attention over the last few years [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 [2] and the trapdoor-knapsack scheme [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 [4]. In addition, it has been shown that if the density of the knapsack is low, where density is loosely defined as the ratio of message text bits to cryptogram bits, then even non-superincreasing knapsacks are insecure [5, 6]. Finally, it should be noted that the inherent message expansion that occurs in a knapsack scheme makes the system difficult to use for authentication. There are ways round this problem [3], but the inherent bijective mapping used in the RSA makes that scheme superior for public-key digital signatures. 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 20 Mbits/s, whereas the RSA is limited to about 50 Kbits/s, 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 [7]. The system bears a resemblance to the Lu–Lee [8] system, but, whereas their cryptosystem is linear and has been shown to be insecure [9], ours is based on the general modular knapsack equation, which to date has not been generally broken.

In describing the algorithm we assume the reader is familiar with public-key cryptography and its terminology. If this is not the case, we refer the reader to one of the many tutorial papers and books available [10, 11, 12].

2 New trapdoor

The general modular knapsack equation is given by

$$S = \sum_{i=1}^{n} a_i \cdot x_i \bmod p \qquad (1)$$

When used for cryptography, the as are the n published knapsack components, p is a published modulus, and the xs are the message bits. In the binary knapsack the xs 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 xs.

Paper 4156E (C3), first received 16th August 1984 and in revised form 31st July 1985

Dr. McAuley and Prof. Goodman were formerly, with the Department of Electronic Engineering, University of Hull, Hull HU6 7RX, United Kingdom. They are now with the Electrical Engineering Department, CALTECH, Pasadena, CA91125, USA

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

$$p = \prod_{i=1}^{n} p_i$$

and where

$$a_j^{(i)} = a_j \bmod p_i$$

is the residue of the jth knapsack component modulo the ith prime. Then, by the Chinese remainder theorem

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

is a bijective mapping. That is, the transformation is one-to-one for all as 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:

$$\boldsymbol{a}' = \begin{matrix} a'_1 \leftrightarrow a_1'^{(1)}, a_1'^{(2)}, \ldots, a_1'^{(n)} \\ a'_2 \leftrightarrow a_2'^{(1)}, a_2'^{(2)}, \ldots, a_2'^{(n)} \\ \vdots \\ a'_n \leftrightarrow a_n'^{(1)}, a_n'^{(2)}, \ldots, a_n'^{(n)} \end{matrix} \tag{2}$$

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

$$a_j = a'_j \cdot W \bmod p \tag{3}$$

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

Now, let

$$p_{i,\,min} \geqslant 2^h \tag{4}$$

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

Let

$$x_{i,\,max} < 2^g \tag{5}$$

that is, the message blocks are g bit numbers.

And let

$$\left(\sum_{j=1}^{n} a_j'^{(i)}\right)_{max} < 2^r \tag{6}$$

that is, the columns of $\boldsymbol{a}'$ 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 $\boldsymbol{x} \rightarrow S$ is injective. To guarantee this we must have

$$h \geqslant r + g \tag{7}$$

which also ensures that modular multiplication is equivalent to matrix multiplication

$$(S'^{(1)}, \ldots, S'^{(n)}) = (x_1, \ldots, x_n) \begin{pmatrix} a_1'^{(1)}, a_1'^{(2)}, \ldots, a_1'^{(n)} \\ \vdots \\ a_n'^{(1)}, a_n'^{(2)}, \ldots, a_n'^{(n)} \end{pmatrix}$$

i.e.

$$\boldsymbol{S}' = \boldsymbol{x} \cdot \boldsymbol{a}'$$

and that the transformation can be inverted (provided the matrix $\boldsymbol{a}'$ is nonsingular) via

$$\boldsymbol{x} = \boldsymbol{S}' \cdot \boldsymbol{a}'^{-1} \tag{8}$$

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

$$S = (x_1 \cdot a_1 + x_2 \cdot a_2 + \cdots + x_n \cdot a_n) \bmod p$$

via eqn. 1. We compute S'

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

and express in modular form using our known factorisation of p

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

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

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

$$\boldsymbol{a}' = \begin{matrix} a'_1 = (3, 1, 1) \leftrightarrow 125174 \\ a'_2 = (1, 5, 3) \leftrightarrow 151664 \\ a'_3 = (2, 1, 2) \leftrightarrow 122509 \end{matrix}$$

Now choose $W = 6553$, which is relatively prime to $p = 65231$. Perform the modular multiplication of eqn. 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 $\boldsymbol{a}'$

$$\boldsymbol{a}'^{-1} = (1/16) \begin{pmatrix} +7 & -1 & -2 \\ +4 & +4 & -8 \\ -9 & -1 & +14 \end{pmatrix}$$

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

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

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

$$\begin{aligned} S' &= 50660 \cdot 2618 \bmod 65231 \\ &= 13257 \bmod 65231 \end{aligned}$$

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

$$\boldsymbol{S}' = (11, \quad 14, \quad 13) \leftrightarrow 13257.$$

From eqn. 8, the receiver computes

$$16 \cdot \boldsymbol{x} = (11, \quad 14, \quad 13) \begin{pmatrix} +7 & -1 & -2 \\ +4 & +4 & -8 \\ -9 & -1 & +14 \end{pmatrix}$$

giving

$x = (1, \quad 2, \quad 3)$ as transmitted.

4 Practical constraints

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

First, consider the security of the trapdoor. In order to ensure that the published p is not factored we set

$$h \geqslant 255 \tag{9}$$

so that the primes are at least 256-bit numbers.

Consider now an attack on the disguising modular multiplication of eqn. 3. Re-arranging eqn. 3 for two particular knapsack components we can form

$$a_k - a'_k \cdot W = 0 \bmod p$$

and

$$a_l - a'_l \cdot W = 0 \bmod p$$

Differencing these two equations we can then form

$$a_k \cdot a'_l - a_l \cdot a'_k = 0 \bmod p \tag{10}$$

and, similarly, for all the residues modulo our secret primes, we find

$$a_k \cdot a_l'^{(i)} - a_l \cdot a_k'^{(i)} = 0 \bmod p_i \tag{11}$$

Given that the a_k and a_l are public, eqn. 11 shows that we need sufficient randomness in the knapsack components in order to prevent the attacker trying all possible pairs $a_l'^{(l)}$ and $a_k'^{(i)}$ and thus breaking the trapdoor. If we assume that any number $1 \geqslant a_j'^{(i)} \geqslant 2^r$ can be chosen to be a knapsack component, then, to prevent the attack, we should set

$$r \geqslant 64 \tag{12}$$

Secondly, let us consider the security of the knapsack. In the past few years there have been rapid advances in solving the basic knapsack problem [5, 6]. Although these attacks have been on the binary knapsack ($g = 1$), the techniques can be extended to cover the general knapsack problem. The choice of g and n will determine the knapsack security.

In order to present a large knapsack problem we set

$$n \cdot g \geqslant 256 \tag{13}$$

Consider now the value of n. This is affected by several conflicting factors. First, 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 lower limit on n. If we use the parameters of eqn. 13 in a binary knapsack, then $g = 1$ and $n = 256$, say. In this case, the least significant bit of the subset sum depends on up to 256 bits of the message. In the general knapsack $k > 1$, and so n is reduced via eqn. 13. Consequently, the involvement of the least significant bit is also reduced.

In order to increase this involvement we can randomly set the last v bits of each message subblock instead of using these for information. We can show the average involvement of the least significant information bit is then given by $n(v + 1)^2/2$. The overall efficiency of the system will then be degraded by a factor v/g. In order to protect the least significant bits we should set

$$\frac{n \cdot (v + 1)^2}{2} \geqslant 128 \tag{14}$$

The final factor affecting n is the size of the public key, this is given by

$$PK = n \cdot (n + 1) \cdot (h + 1) \text{ bits} \tag{15}$$

As this varies with the square of n, we should aim to keep n as small as possible, as well as h. From eqn. 9, we can set $h = 255$. If we choose $n = 7$, then the size of the public key is 14336 bits by eqn. 15, which compares favourably with the Merkle–Hellman scheme.

Consider now the value of r, which also determines the value of g via eqn. 7. A recent method of attacking knapsacks is given in Reference 5 and is based on forming a lattice of rank n. If the density of the knapsack is low, then this method can successfully break any binary knapsack. In particular, if the density D is such that

$$D = \frac{n}{\log_2 (\max a_j)} < \frac{1}{\log_2 n}$$

the method is successful. However, for the general knapsack, it is more difficult to find a suitable lattice. In our case the density is given by

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

if we assume the primes are all $h + 1$ bit numbers. So, 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 equality ($h = r + g$), so that

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

Now, to maximise D, we must keep r small. From eqn. 12 we therefore set $r = 64$, giving $g = 255 - 64 = 191$. The size of the message block is then $n \cdot g = 1337$ bits, which certainly satisfies eqn. 13. The efficiency of the system is given by

$$E = \frac{g - v}{h + 1} \tag{16}$$

If we choose $v = 6$ in order to satisfy eqn. 14, then eqn. 16 gives $E = 0.72$.

The final system parameters are then $n = 7$, $r = 64$, $g = 191$, $h = 255$ and $v = 6$. This gives a density $D = 0.75$, a public-key size $PK = 14336$ bits and an overall efficiency of $E = 0.72$.

5 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 0.75 and a block size of 1337 bits. 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 14 Kbits, is not excessive when compared with 80 K bits for the Merkle–Hellman scheme and 1 Kbits for the RSA. It may be possible to attack the trapdoor information via the methods in Reference 5, 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.

6 References

1 DIFFIE, W., and HELLMAN, M.: 'New directions in cryptography', *IEEE Trans.*, 1976, **IT-22**, pp. 644–654

2 RIVEST, R., SHAMIR, A., and ADELMAN, L.: 'On digital signatures and public key cryptosystems', *Comm. ACM*, 1978, **21**, (2), pp. 120–126

3 MERKLE, R.C., and HELLMAN, M.E.: 'Hiding information and signatures in trapdoor knapsacks', *IEEE Trans.*, 1978, **IT-24**, pp. 525–530

4 DESMET, Y., VANDEWALLE, J., and GOVAERTS, R.: 'A critical analysis of the security of knapsack public key cryptosystems'. IEEE Symposium on Information Theory, Les Arcs, France, June 1982

5 BRICKELL, E.F.: 'Solving low density knapsacks', Sandia National Laboratories, Albuquerque, New Mexico, USA 1983, p. 13

6 LAGARIAS, J.C., and ODLYZKO, A.M.: 'Solving low-density subset sum problems', Bell Laboratories, Murray Hill, New Jersey, USA, 1983, p. 38

7 KNUTH, D.E.: 'The art of computer programming, Vol. 1', *in* 'Fundamental algorithms' (Addison-Wesley, 1968)

8 LU, S.C., and LEE, L.N.: 'A simple and effective public key cryptosystem', *Comsat Tech. Rev.*, 1979, **9**, pp. 15–24

9 GOETHALS, J.M., and COUVREUR, C.: 'A cryptanalytic attack on the Lu-Lee public key cryptosystem', *Phillips J. Res.*, 1980, **35**, pp. 301–306

10 DENNING, D.E.: 'Cryptography and data security' (Addison-Wesley, 1982)

11 WILLETT, M.: 'A tutorial on public key cryptography', *Comput. & Secur.*, 1982, **1**, pp. 72–79

12 SIMMONS, G.J.: 'Cryptography: the mathematics of secure communication', *Math. Intelligencier*, 1979, **1**, pp. 233–246