A new trapdoor in knapsacks

Valtteri Niemi
Mathematics Department
University of Turku
20500 Turku, Finland

Abstract. *A public key scheme with trapdoor based on a group of modular knapsacks is proposed. In parallel architecture encryption and decryption are very fast.*

1 Introduction

Recently Adi Shamir [4] proposed a new identification scheme that possesses two nice characteristic features. First, he got rid of huge number arithmetics thus making it easier to implement a smart card identification system. Secondly, the security of the new scheme does not depend on the difficulty of factoring.

In this paper we present a new public key cryptosystem based on the same two features as Shamir's identification scheme. We do not obtain as low levels of time and space complexity as in [4] (which is quite understandable since we must build a trapdoor). Nevertheless, we may use 8 bit numbers and at least in parallel architecture the operations can be carried out very fast.

The underlying difficult problem we try to imitate consists of a group of modular knapsack type equations. It is NP-complete in the strong sense which means that it is possible to use small numbers as coefficients in the equations.

Our trapdoor is built roughly as follows. Let us consider square matrices whose elements are very small numbers. We disguise them by matrix multiplication with an arbitrary square matrix. All operations are carried out in a finite field of a prime order. Note that if we originally have only one "small" square matrix this process causes no limitations on the disguised matrix obtained after multiplication (provided the original matrix has an inverse).

In our scheme, we begin with two "small" matrices and use the two disguised versions as a public key. (They constitute the coefficients in knapsack type equations.) The security of the trapdoor is thus based on the difficulty of determining the multiplier matrix after two repetitions of the disguising procedure.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 405-411, 1991.

2 The underlying hard problem

Let us consider two $n \times n$-square matrices A and B over a finite field $\mathbf{Z}_p$. Combine these matrices to one $n \times 2n$-matrix $E = (A \mid B)$. Let x be a column vector (i.e. $2n \times 1$-matrix) of 0's and 1's. Now we obtain an $n \times 1$-matrix c as a matrix product : $c = Ex$.

Modular knapsack group problem :
Given : an $n \times 2n$-matrix E, an n-vector c and a prime p.
Find : a $2n$-vector x whose elements belong to the set $\{0, 1\}$ satisfying the equation $Ex \equiv c \pmod{p}$.

A quite straight-forward transformation from the EXACT 3-COVER problem (see [1]) shows that our problem is NP-complete even in the case where the elements of E and c are also 0's and 1's. Hence, the problem is NP-complete in the strong sense.

3 The trapdoor

Let us first define a notion of an *absolute value* in the field $\mathbf{Z}_p$: An absolute value $|a|$ of $a \in \mathbf{Z}_p$ is the minimum of the least *positive* remainders (modulo p) of the two integers a and $-a$.

Example. In $\mathbf{Z}_{19}$ $|3| = |16| = 3, |-8| = |8| = |11| = |30| = 8$ etc.

In general, $0 \leq |a| \leq \frac{p}{2}$. We say that a is k-small if $|a| \leq k$, and a is k-large if $|a| \geq \frac{p}{2} - k$. In the sequel, we speak shortly of small and large numbers thus leaving k unfixed. However, we make a general unrigorous assumption that k is relatively small compared with the moduli p. (The exact choice of k is discussed in the next section.)

Let us now fix $n \times n$-square matrices C,D,S of *small* numbers and an arbitrary $n \times n$-matrix R. Furthermore, fix a *diagonal* $n \times n$-matrix Δ of *large* numbers. Compute the two square matrices

$$A = R^{-1}(\Delta - SC) \quad \text{and} \quad B = -R^{-1}SD \tag{1}$$

(which exist provided R has a full rank.)

Now the following matrix identity holds :

$$\begin{pmatrix} R & S \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} \Delta & 0 \end{pmatrix}.$$

Our cryptosystem can now be defined :

Public key : Matrix $E = (\ A \quad B\ )$.

Private key : Matrix R. (Matrices C, D, S and Δ are also private but they are not needed after the initial construction.)

Messages : $2n$-bit vectors x.

Encryption : an n-vector $c = Ex$.

Decryption : Compute an n-vector $l = Rc$. Now the first half of x can be found by the following rule :

$$\begin{cases} \text{If } l_i \text{ is small then } x_i = 0 \\ \text{If } l_i \text{ is large then } x_i = 1 \end{cases} \qquad (i = 1, ..., n).$$

The other half of x can be decrypted by elementary linear algebra (n equations, n variables).

Our decryption rule is valid by the following observations. First,

$$(\ R \quad S\ )\begin{pmatrix} A & B \\ C & D \end{pmatrix}(x) = (\ \Delta \quad 0\ )(x) = \begin{pmatrix} \Delta_1 x_1 \\ \vdots \\ \Delta_n x_n \end{pmatrix},$$

where Δ_i:s are Δ:s (large) diagonal elements.

On the other hand,

$$(\ R \quad S\ )\begin{pmatrix} A & B \\ C & D \end{pmatrix}(x) = (\ R \quad S\ )\begin{pmatrix} c \\ \alpha \end{pmatrix},$$

where α is a small n-vector, since C, D and x consist of small numbers (of course, the parameter k in the definition of smallness must be increased in the case of α).

Furthermore,

$$(\ R \quad S\ )\begin{pmatrix} c \\ \alpha \end{pmatrix} = Rc + S\alpha = l + S\alpha,$$

where $S\alpha$ is a small n-vector as S and α consist of small numbers only.

Thus, $\Delta_i x_i = l_i + (S\alpha)_i$ and since the last term is small, l_i determines whether the right-hand side is large or small. Similarly, x_i determines whether the left-hand side is large or small.

A toy example. Let us fix $n = 5, k = 1$ and $p = 23$. We choose matrices

C,D (whose elements are 1-small) and R at random :

$$C = \begin{pmatrix} 0 & 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}, D = \begin{pmatrix} 0 & 1 & 1 & 0 & -1 \\ 1 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$R = \begin{pmatrix} 15 & 14 & 14 & 11 & 21 \\ 6 & 17 & 6 & 17 & 4 \\ 13 & 15 & 11 & 7 & 13 \\ 19 & 18 & 19 & 7 & 3 \\ 18 & 13 & 14 & 8 & 8 \end{pmatrix}.$$

Also, we choose $S = I$ (the identity matrix) and $\Delta = 11 \cdot I$. Now

$$R^{-1} = \begin{pmatrix} -7 & 3 & 17 & 4 & 5 \\ 1 & 1 & 4 & 11 & -8 \\ 11 & -5 & -1 & 2 & 9 \\ 17 & 4 & 13 & 9 & 18 \\ 21 & 5 & 13 & -2 & -6 \end{pmatrix}$$

and we may calculate the public key matrices by (1) :

$$A = \begin{pmatrix} 5 & 11 & 3 & 6 & 20 \\ -4 & 15 & 11 & 9 & 13 \\ 4 & 6 & -3 & 3 & 9 \\ 3 & 13 & 12 & 5 & 13 \\ 14 & 2 & 6 & -4 & 7 \end{pmatrix}, B = \begin{pmatrix} 9 & 8 & 10 & 6 & 13 \\ 11 & 9 & 0 & -4 & -6 \\ 18 & -4 & 7 & 1 & 1 \\ -9 & 11 & 10 & 10 & 0 \\ 14 & -5 & 7 & 10 & 16 \end{pmatrix}.$$

The message

$$x = (\ 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1\ )$$

is encrypted as

$$c = (\ A \quad B\ ) \cdot x^T = (\ -9 \quad 21 \quad -6 \quad 9 \quad 13\ )^T.$$

To decrypt we first calculate

$$l = Rc = (\ 10 \quad 12 \quad 19 \quad 11 \quad 19\ )^T = (\ 10 \quad 12 \quad -4 \quad 11 \quad -4\ )^T$$

from which we can derive the first 5 bits of the message: first, second and fourth element are large, thus corresponding to 1's, while third and fifth element are (comparatively) small, corresponding to 0's. We skip here the second part of decyption (that uses methods of linear algebra).

4 Observations on security and complexity

As already noted in the previous section, it is sufficient to decrypt only half of the cryptotext bits, since the other half can be determined easily by the public key only. Of course, the cryptanalyst is also able to complete the decryption if she already knows half of the plaintext. Also, it is possible to derive dependencies between plaintext bits solely from the public information. For these reasons it is recommended to combine each n-bit plaintext block with a random padding of n bits.

From (1) in the previous section we see that the matrices A and B are obtained from two special-type matrices (i.e. $\Delta - SC$ and $-SD$) by multiplying them with the *same* matrix. In principle, this relation between A and B gives us a starting point for cryptanalysis. The multiplier R^{-1} can be eliminated, leading to the equation

$$-SD = (\Delta - SC)A^{-1}B.$$

Hence, we should be able to multiply the known matrix $A^{-1}B$ by some matrix whose nondiagonal elements are small and diagonal ones are large and the result of this operation should be a small matrix. It is easy to see that if we can fulfil these conditions, a suitable decryption matrix, say R', can be found.

This approach also shows that from the point of security the choices of S and Δ are quite unessential. We may, for instance, let S to be an identity matrix I and Δ to be $\lfloor \frac{p}{2} \rfloor \cdot I$. (Hence, S and Δ could be universal entities.) Now the critical requirement of the previous section, i.e. that $S\alpha$ should be small, can be stated exactly in the form of inequality :

$$|\alpha| < \lfloor \frac{p}{4} \rfloor.$$

On the other hand, since $\alpha = (\ C \quad D\ ) \cdot (x)$, it follows from the triangle inequality (which clearly is valid also for this definition of absolute value) that

$$|\alpha| \leq 2n \cdot \max\{|c| : c \in C \cup D\} = 2nk \tag{2}$$

(where k refers to the definition of k-smallness). Hence, the critical demand can be restated as

$$nk < \frac{p}{8} \tag{3}$$

To exclude exhaustive search attacks we should choose n sufficiently large, e.g. $n = 100$ seems to be suitable. If $k = 1$, which means that the elements of C and D must be chosen from the set $\{-1, 0, 1\}$, the condition (3) gives a lower bound for the moduli p : $p > 800$. This would mean that 8 bit numbers are slightly too small for our purposes. However, in practice the value of $|\alpha|$ is considerably smaller than the theoretical upper bound $2nk$ derived in (2).

Indeed, the value $2nk$ could be reached only in the extreme case where all elements of C, D and x are equal to 1. Since the choice of C and D is free within the set $\{-1, 0, 1\}$, we can easily reduce even the theoretical upper bound to one third or even more. This gives us the possibility of using, e.g., a moduli $p = 251$ that is a 8 bit number.

In fact, a false decryption of some bits due to too large value of $|\alpha|$ does not cause serious problems, seen as follows. Let us suppose the receiver has decrypted a cryptogram and she has a proposal for the correct plaintext. She can easily check whether the proposal is valid by the encryption mechanism. If the result of checking is negative she must determine which bits are wrong. Fortunately, the receiver can use her proposal to calculate estimates for the $|\alpha|$-values. Bits corresponding to largest $|\alpha|$-values are best candidates as false ones, and the receiver can correct her guess. Of course, the process converges only if the portion of falsely decrypted bits is small.

Another question is the choice of k. If one does not trust on too small elements in C and D but rather chooses, e.g., $k = 128$ the size of the moduli respectively extends to 16 bits etc. On the other hand, it does not seem to be very likely that the security of the system would depend too heavily on the size of k.

In the case of $n = 100, k = 1$ we must, in average, execute 10 000 single-byte additions to encrypt a padded 100-bit message. Decryption takes the same number of single-byte multiplications. On the other hand, the mechanism is particularly suitable for parallel computers, since only matrix operations are needed. For instance, a special hardware with 100 processors needs only one single-byte addition to encrypt one bit of plaintext and one single-byte multiplication per decrypted bit. With more processors involved the operations are even faster.

The keys are, unfortunately, quite large. The public key consists of 20 kilobytes in the same case as above, while the private key needs 10 kB. The latter one is basically a random matrix, hence it can be stored in a form of a pseudorandom function but the same idea does not suit for the public matrix. In general it can be said that as regards time and space requirements this new system is in the same class as some known systems based on error-correcting codes (see e.g. [3] and [2].)

5 Some variations

Perhaps the most immediate variations of the basic scheme are found by changing the underlying field structure. For example, we could choose elements of R^{-1}, Δ, S, C and D from $\mathbf{Z}$ (which means that also A and B are $\mathbf{Z}$-matrices). In this case, the decryption matrix R will be a rational matrix, and obviously both keys will be larger than in the modular version. Finite fields of order p^k do not seem very suitable, since the ordering in the field is crucial to the scheme.

Another possible variation is to change the form of the matrices involved. Let A and B be general $n \times m$-matrices instead of square ones. Similarly, C and D are $n \times m$-matrices and, respectively, R and S are $m \times n$-matrices. Further, Δ is still a square (of order m). The length of plaintext blocks is $2m$, while cryptotext blocks are still n-vectors. This variant means that R cannot be chosen completely randomly in the case $m > n$.

The following variation deals with the problem of large keys. Recall that the private key R was chosen at random and the public key matrices were calculated by (1). We could as well choose A (or B) at random and calculate R and B (or R and A). Then half of the public key is random and could be stored in pseudorandom form, thus reducing the need of storage essentially by half.

Our last variation is in fact an addition to the disguising procedure. We can try to improve the security of the system by a standard way of permuting the columns of the encryption matrix E. As a result there is no trivial way of separating matrices A and B, hence the starting point of cryptanalysis must be based on the properties of single columns.

Unfortunately, we have no variants for the purpose of signatures. As usual in knapsack-type systems, the encryption function is not surjective.

References

[1] Garey, M. R. and Johnson, D. S., *Computers and intractability : A guide to the theory of NP-completeness*, 1979.

[2] MacEliece, R. J., A public-key cryptosystem based on algebraic coding theory, *DSN Progress Rep. 42-44, Jet Propulsion Laboratory*, 114-116, 1978.

[3] Niederreiter, H., Knapsack-type cryptosystems and algebraic coding theory, *Problems of Control and Information Theory*, **15**, 159-166, 1986.

[4] Shamir, A., An Efficient Identification Scheme Based on Permuted Kernels (extended abstract), 1989.