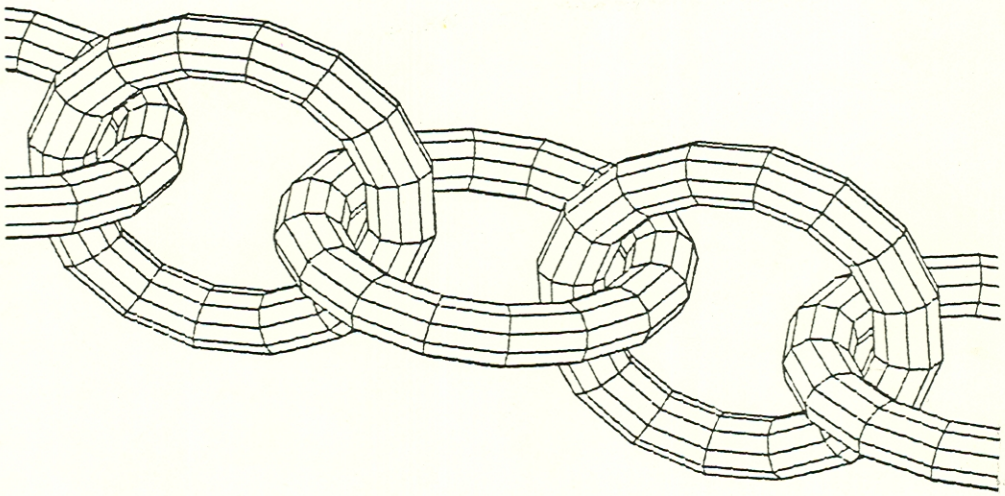
MIT/LCS/TM-230

EMBEDDING CRYPTOGRAPHIC TRAPDOORS IN ARBITRARY KNAPSACK SYSTEMS



Adi Shamir

September 1982

EMBEDDING CRYPTOGRAPHIC TRAPDOORS IN ARBITRARY KNAPSACK SYSTEMS

by

Adi Shamir
Applied Mathematics
The Weizmann Institute
Rehovot, Israel

Abstract

In this paper we show that after sufficiently many modular multiplications, any knapsack system becomes a trapdoor system that can be used in public-key cryptography.

Key Words

Cryptography, Merkle-Hellman cryptosystems, knapsack problems

Cover design by John G. Aspinall.

This research was partially supported by NSF grant no. MCS-8006938.

I. Introduction

All the knapsack-based public-key cryptosystem proposed so far (e.g., Merkle and Hellman [1978] and Shamir and Zippel [1980]) are based on the following paradigm:

1. Pick an easy knapsack system;
2. Scramble its elements by iterated modular multiplications;
3. Publish the resultant randomly-looking system.

In this paper we show how to replace part 1 by:

1'. Pick an arbitrary knapsack system.

The initial knapsack system need not be one-to-one and its elements can be of arbitrary sizes. All the known knapsack cryptosystems are special cases of the new "random knapsack cryptosystem," and thus any successful cryptanalytic attack against it will also break the other cryptosystems (but not necessarily vice versa). In addition, the fact that the initial knapsack system has no structure whatsoever implies that the cryptanalyst cannot expose the secret trapdoor by looking for this structure (as was done, for example, in Shamir [1982]), and his only hope is to attack the scrambling mechanism itself. While the modular multiplication technique is the only known method for scrambling knapsacks without changing their solutions, it is conceivable that other methods will be found in the future. The idea presented in this paper can be used with any iterative scrambling technique, as long as all the intermediate knapsack problems have the same solutions and their entries are randomly-looking.

II. The Random Knapsack Cryptosystem

Let $a_1^1,\ldots,a_n^1$ be an arbitrary initial knapsack system, and let $a_1^j,\ldots,a_n^j$ be the system obtained after $j - 1$ modular multiplications. We denote the j-th modulus by M^j and the j-th multiplier by W^j, and define:

$$a_i^{j+1} = W^j \cdot a_i^j \pmod{M^j}$$

The values of M^j and W^j are arbitrary, as long as they are relatively prime and

$$M^j > \sum_{i=1}^{n} a_i^j$$

These modular multiplications are iterated $n - 1$ times, and the resultant system $a_1^n,\ldots,a_n^n$ is published as the encryption key.

Given a binary cleartext $x_1\ldots x_n$ (in which each x_i is 0 or 1), the sender computes a message-dependent partial sum of $a_1^n,\ldots,a_n^n$:

$$\sum_{i=1}^{n} x_i a_i^n = b^n$$

and sends the ciphertext b^n to the reciever over the (insecure) communication channel.

To decrypt b^n, the reciever multiplies the equation by the inverse of W^{n-1} modulo M^{n-1} (the inverse exists since W^{n-1} and M^{n-1} are relatively prime). Each a_i^n is transformed back into a_i^{n-1}, and the b^n is changed to a new value b^{n-1}:

$$\sum_{i=1}^{n} x_i a_i^{n-1} = b^{n-1} \pmod{M^{n-1}}$$

Since each x_i is at most 1 and M^{n-1} is larger than the sum of all the a_i^{n-1}, this equation holds even when the (mod M^{n-1}) clause is deleted:

$$\sum_{i=1}^{n} x_i a_i^{n-1} = b^{n-1}$$

Continuing in this way, the reciever can gather n non-modular equations in n unknowns of the form:

$$\sum_{i=1}^{n} x_i a_i^{j} = b^{j} \qquad j = 1,\ldots,n$$

The a_i^j coefficients in these equations are randomly looking, and when they are large enough, the system of equations is almost certainly non-singular. If this is the case, the reciever can solve the equations and find $x_1,\ldots,x_n$ without relying on the easy solvability of the initial knapsack system $a_1^1,\ldots,a_n^1$.

Instead of solving the equations over the rationals, the reciever can reduce the equations mod 2 and solve them over GF(2). The reduced equations contain only the least significant bits of a_i^j and b^j, and they are much easier to store and manipulate. However, for random binary matrices, the probability of non-signularity is only 0.3, and thus a few random keys have to be generated before a useful one is found. The inverse of the matrix of least significant a_i^j bits can be precomputed during the key generation phase, and thus the decryption procedure consists of collecting the least significant bits of the b^j numbers and multiplying this vector by a fixed

n x n binary matrix.

Example

Consider the initial knapsack system 3, 8, 11. If we choose $M^1 = 25$ and $W^1 = 14$, we get the transformed system 17, 12, 4. By using $M^2 = 37$ and $W^2 = 17$, we get the doubly transformed system 27, 19, 30. For any ciphertext b^3, the system of equations is:

$$x_1 \cdot 3 + x_2 \cdot 8 + x_3 \cdot 11 = b^1$$

$$x_1 \cdot 17 + x_2 \cdot 12 + x_3 \cdot 4 = b^2$$

$$x_1 \cdot 27 + x_2 \cdot 19 + x_3 \cdot 30 = b^3$$

This system can be soved over the rationals, but it is simpler to reduce it mod 2:

$$x_1 \cdot 1 + x_2 \cdot 0 + x_3 \cdot 1 = \tilde{b}^1$$

$$x_1 \cdot 1 + x_2 \cdot 0 + x_3 \cdot 0 = \tilde{b}^2$$

$$x_1 \cdot 1 + x_2 \cdot 1 + x_3 \cdot 0 = \tilde{b}^3$$

where $\tilde{b}^j$ is the least significant bit of b^j.

Since over GF(2)

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

for any vector of $\tilde{b}^j$ we can compute the x_i as:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \end{bmatrix}$$

Note that this system of equations is non-singular even though the original knapsack system 3, 8, 11 is not one-to-one (11 can be represented in two different ways as the sum of a subset of the elements).

□

To be secure, any knapsack-based cryptosystem must have at least 100 unknown x_i bits, and thus the decryption matrix contains at least 10,000 bits. There is an easy way to control the structure of this matrix and to transform it into any particular matrix we choose. Let $M^1,\ldots,M^{n-1}$ be odd numbers. After multiplying $a_1^j,\ldots,a_n^j$ by W^j and reducing them mod M^j, we can add M^j selectively to any subset of the results, since these extra occurrences of M^j will be eliminated during the decryption phase by the inverse multiplications mod M^j. Since M^j is odd, the least significant bit of a_i^{j+1} is flipped when M^j is added to it, and thus we can independently select all the entries of the matrix. In particular, we can choose a knapsack system for which the coefficient matrix is the identity matrix; the cleartext in this case is simply the sequence of least significant bits of the intermediate b^j numbers! However, this technique introduces a small amount of known structure into the

intermediate knapsack systems, and thus to get the highest possible security it seems advisable to leave the matrix random.

Example (continued)

To get a 3 x 3 identity matrix, we augment our initial knapsack system to 3, 8, 12. With the same $M^1 = 25$ and $W^1 = 14$, we get the transformed sequence 17, 12, 18. We change this sequence to 42, 37, 18 by adding 25 to 17 and to 12. The sum of the new sequence is 97, and thus we cannot use the modulus $M^2 = 37$ from the original example. Choosing $M^2 = 101$, $W^2 = 23$, we get the doubly transformed sequence 57, 43, 10. By adding 101 to all of them, we get the sequence 158, 144, 111. The three sequences

(1) 3, 8, 12

(2) 42, 37, 18

(3) 158, 144, 111

have the desired structure of least significant bits, and the reader can easily verify that the j-th sequence can be obtained from the (j + 1)-st sequence by inverse modular multiplications. Note that to prevent exposure of the last cleartext bit, it is necessary to add a final scrambling stage in order to make the least significant bits of the published key randomly looking. □

III. Conclusions

In this paper we have shown that the modular multiplication technique can embed its own trapdoor in knapsack systems, and thus it is not necessary to use easy-to-solve initial systems.

One of the corollaries of this observation is that Merkle-Hellman cryptosystems may be weakened rather than strengthened by too many modular multiplications, since they introduce new unintentional trapdoors into the knapsack systems. In particular, it makes no sense to scramble an n-element Merkle-Hellman cryptosystem more than n times, since both the reciever and the cryptanalyst can easily compute the cleartext by unscrambling only the last n iterations. On the other hand, too few iterations also seem to weaken the cryptosystem, and thus the question of the optimal number of iterations requires careful study.

The new "random knapsack cryptosystem" proposed in this paper is as secure as any other multi-iteration knapsack-based cryptosystem, since it makes no assumptions about the initial system. However, it shares with all the other cryptosystems the potential weakness that each inverse modular multiplication is known to reduce the size of all the knapsack elements by at least $\log n$ bits. Information theoretically, this is a very strong clue, but it is an open problem whether the cryptanalyst can use it in a computationally efficient manner.

Acknowledgements: I would like to thank Ron Rivest for the many useful and enjoyable discussions we had on knapsack cryptosystems.