KNAPSACK PUBLIC KEY CRYPTOSYSTEMS AND

DIOPHANTINE APPROXIMATION

(Extended Abstract)

J. C. Lagarias

AT&T Bell Laboratories

Murray Hill, New Jersey

0. Abstract

This paper presents and analyzes cryptanalytic attacks on knapsack public key cryptosystems that are based on ideas from Diophantine approximation. Shamir's attack on the basic Merkle-Hellman knapsack cryptosystem is shown to depend on the existence of ''unusually good'' simultaneous Diophantine approximations to a vector constructed from the public key. This aspect of Shamir's attack carries over to multiply iterated knapsack cryptosystems: there are ''unusually good'' simultaneous Diophantine approximations to an analogous vector constructed from the public key. These ''unusually good'' simultaneous Diophantine approximations can be used to break multiply iterated knapsack cryptosystems provided one can solve a certain nonlinear Diophantine approximation problem. This nonlinear problem is solved in the simplest case and then used to give a new cryptanalytic attack on doubly iterated knapsack cryptosystems.

1. Introduction

In 1978 Merkle and Hellman [14] proposed a public key cryptosystem based on the idea of a ''trapdoor'' knapsack. In their cryptosystem the public information is a set of knapsack weights $\{a_i : 1 \le i \le n\}$ which are positive integers. Encryption is done by taking the plaintext, assumed to be given in blocks $(x_1, \ldots, x_n)$ of n binary bits, and sending as ciphertext for the block the integer E given by

$$\sum_{i=1}^{n} a_i x_i = E \, . \tag{1.1}$$

The knapsack weights $\{a_i: 1 \le i \le n\}$ are chosen so that the equation (1.1) has at most one solution with all $x_i = 0$ or 1, so that the plaintext is uniquely recoverable from (1.1). Since the problem of finding a solution $(x_1, \ldots, x_n)$ to (1.1) with all $x_i = 0$ or 1 given *arbitrary* weights $\{a_i: 1 \le i \le n\}$ and E is known to be *NP*-hard, it is necessary to choose weights $\{a_i: 1 \le i \le n\}$ with some kind of special structure in order that decryption can be performed easily. One also wants this special structure to be hidden by a ‘‘trapdoor’’ so that it is concealed from all but the authorized receiver. Merkle and Hellman’s special structure consists of *superincreasing* knapsack weights $\{s_i: 1 \le i \le n\}$, which are weights satisfying the conditions

$$\sum_{i=1}^{j} s_i < s_{j+1} \quad \text{for} \quad 1 \le j \le n-1 \ .$$

Decryption can be carried out in linear time for such sequences. Merkle and Hellman use modular multiplication as the ‘‘trapdoor’’. The *basic Merkle-Hellman cryptosystem* has as trapdoor information a superincreasing sequence $\{s_i: 1 \le i \le n\}$, a modulus M and a multiplier W satisfying $1 \le W < M$ and $(W, M) = 1$. The public weights $\{a_i: 1 \le i \le n\}$ are determined by

$$W a_i \equiv s_i \pmod{M} \ , \tag{1.2}$$

with $1 \le a_i < M$. The modulus M is required to satisfy the *size condition*

$$M > \sum_{i=1}^{n} s_i \ , \tag{1.3}$$

to allow easy decryption. To decrypt, we multiply (1.1) by W to obtain

$$\sum_{i=1}^{n} s_i x_i = WE - ➾ M \tag{1.4}$$

for some integer ➾. The size condition (1.3) ensures that if E_1 is determined by

$$E_1 \equiv WE \pmod{M_1} \tag{1.5}$$

and $1 \le E_1 < M$ then in fact $E_1 = WE -$ ➾M so that we may directly recover the plaintext by solving the equation

$$\sum_{i=1}^{n} s_i x_i = E_1; \ \ all \ \ x_i = 0 \ \text{ or } \ 1 \ .$$

This is easily done since the weights $\{s_i: 1 \le i \le n\}$ are superincreasing. Merkle and Hellman went on to propose using repeated modular multiplications to improve the security of the trapdoor. The *m-times iterated knapsack cryptosystem* has as trapdoor information a superincreasing sequence $\{s_i: 1 \le i \le n\}$ and set of m multiplier-modulus pairs $\{(W_j, M_j): 1 \le j \le m\}$. We let

$$W_1 a_i^{(1)} \equiv s_i \pmod{M_1} \ , \quad 1 \le i \le n \ ,$$

with $1 \le a_i^{(1)} < M_1$ and

$$W_j a_i^{(j)} \equiv a_i^{(j-1)} \pmod{M_j} \ ; \ \ 1 \le i \le n \ , \tag{1.6}$$

with $0 \le a_i^{(j)} < M_j$ for $2 \le j \le m$. The moduli are required to satisfy the *size conditions*

$$M_1 > \sum_{i=1}^{n} s_i$$

and

$$M_j > \sum_{i=1}^{n} a_i^{(j-1)} \; ; \quad 2 \le j < m ,$$

in order to allow easy decryption. The public knapsack weights are $\{a_i^{(m)} : \; 1 \le i \le n\}$.

A basic parameter measuring the performance of knapsack cryptosystems is the information rate R, which roughly speaking measures the ratio

$$R \cong \frac{\#\ \text{plaintext bits}}{\#\ \text{ciphertext bits}} \; . \tag{1.7}$$

This quantity on the right side of (1.7) is not precisely defined, because the number of ciphertext bits may depend on the cleartext; therefore we formally define the *information rate R* for a knapsack cryptosystem with weights $\{a_i : \; 1 \le i \le n\}$ by

$$R = \frac{n}{\log_2 \left(\sum_{i=1}^{n} a_i \right)} \; . \tag{1.8}$$

Note that $0 < R \le 1$ holds, because $\sum_{i=1}^{n} a_i \ge 2^n$ since the 2^n possible ciphertexts are all distinct.

In 1982 Adi Shamir [16] made a fundamental cryptanalytic breakthrough, finding a polynomial time attack on ‘‘almost all’’ basic Merkle-Hellman cryptosystems, when the information rate R is held fixed and the block size $n \to \infty$. Since then others have found attacks on other knapsack cryptosystems, notably L. Adleman, who found an attack to break the Graham-Shamir knapsack scheme [2]. Adleman ([1], [2]) also has proposed an attack on iterated knapsack cryptosystems, but at present it is not clear to what extent this attack succeeds, see [4].

This paper presents and analyzes cryptanalytic attacks on knapsack cryptosystems based on ideas from Diophantine approximation. Diophantine approximation is a branch of number theory which in its most special form studies the approximation properties of real numbers θ by rational numbers $\frac{p}{Q}$, and in its most general form deals with the simultaneous close approximation to zero of a set of linear forms at integer lattice points. This paper shows that public weights $\{a_i : \; 1 \le i \le n\}$ formed by concealing a superincreasing sequence $\{s_i : \; 1 \le i \le n\}$ by repeated modular multiplications have a fundamental cryptographic weakness, which is that the public weights have ‘‘unusual’’ simultaneous Diophantine approximation properties. In particular we show that Shamir’s attack on the basic Merkle-Hellman cryptosystem may be viewed schematically as:

(i) Extracting from the public knapsack weights a Diophantine approximation problem having at least one ‘‘unusually good’’ approximation.

(ii) Recovering the particular ''unusually good'' approximation corresponding to the original encryption using an algorithm to find good simultaneous Diophantine approximations.

(iii) Completing the attack using the ''unusually good'' approximation recovered.

We go on to show (Theorem D) that conditions (i) and (ii) still hold for multiply iterated knapsack cryptosystems. Stage (iii) leads to a non-linear Diophantine approximation problem which I do not know how to solve in general. However there is a special trick which apparently allows the completion of stage (iii) for doubly iterated knapsack cryptosystems (see Section 7).

The outline of this paper is as follows. In section 2 we discuss simultaneous Diophantine approximation and define more precisely what we mean by an ''unusually good'' simultaneous Diophantine approximation. Section 3 then shows that Shamir's attack is based on ''unusually good'' simultaneous Diophantine approximations and relates their goodness of approximation to the information rate R. Section 4 shows that iterated knapsacks also have ''unusually good'' approximations and relates their goodness of approximation to the information rate R. Section 5 discusses how the problem of recovering ''unusually good'' simultaneous Diophantine approximations may be set up as a problem of finding a short vector in an integral lattice. Section 6 formulates a non-linear Diophantine approximation problem arising from stage (iii) for iterated knapsacks, and finally Section 7 outlines an attack on doubly-iterated knapsacks.

The relation of Diophantine approximation to the cryptanalysis of knapsack cryptosystems extends beyond the topics discussed here. For example it underlies the attacks of Brickell [3] and Lagarias-Odlyzko [11] on low-density knapsack cryptosystems, Adleman's attack on the Graham-Shamir knapsack cryptosystem [2], and Odlyzko's attack on multiplicative knapsack schemes [15]. Diophantine approximation methods can also be used to rigorously analyze the performance of Shamir's attack ([9],[10]). See also the survey [6].

This extended abstract states results without proof. Detailed proofs will appear in a subsequent paper.

2. Simultaneous Diophantine Approximation

Simultaneous Diophantine approximation is the study of the approximation properties of vectors $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$ of n real numbers by vectors $\boldsymbol{\xi} = (\frac{p_1}{Q}, \frac{p_2}{Q}, \ldots, \frac{p_n}{Q})$ of n rational numbers having the same denominator. The goodness of an approximation to a vector $\boldsymbol{\theta}$ is measured in terms of the denominator Q. We use as a measure of approximation with rationals with denominator Q the function

$$\{\{Q\boldsymbol{\theta}\}\} = \max_{1 \le i \le n} \left(\min_{\mathbf{p} \in \mathbf{Z}^n} |Q\theta_i - p_i| \right) , \tag{3.1}$$

i.e. we use the sup norm $\|\cdot\|_s$ where

$$\|\boldsymbol{\theta}\|_s = \max_{1 \le i \le n} |\theta_i| .$$

The following well-known result summarizes how well vectors $\boldsymbol{\theta}$ in $\mathbf{R}^n$ can be simultaneously approximated.

Proposition 2.1

(1) *For every* $\boldsymbol{\theta}$ *in* $\mathbf{R}^n$ *there are infinitely many positive integer solutions to*

$$\{\{Q\boldsymbol{\theta}\}\} \le Q^{-\frac{1}{n}} .$$

(2) *For any fixed* $\varepsilon > 0$ *the set of* $\boldsymbol{\theta}$ *in* $\mathbf{R}^n$ *which have infinitely many integer solutions* Q *to*

$$\{\{Q\boldsymbol{\theta}\}\} \le Q^{-\frac{1}{n} - \varepsilon}$$

has Lebesgue measure zero.

However the simultaneous Diophantine approximation problems we consider do not involve approximations to an arbitrary real vector $\boldsymbol{\theta}$ in $\mathbf{R}^n$, but instead involve approximating a vector $\boldsymbol{\alpha}$ of rationals

$$\boldsymbol{\alpha} = (\frac{a_1}{A}, \frac{a_2}{A}, \ldots, \frac{a_n}{A})$$

with denominator A with a vector $\boldsymbol{\xi} = (\frac{p_1}{Q}, \ldots, \frac{p_n}{Q})$ of rationals with a smaller denominator Q. In this case we use a slightly different measure of goodness of approximation than (3.1) which involves the denominator A of the vector $\boldsymbol{\alpha}$ being approximated. We say that a denominator Q (or a vector $\boldsymbol{\xi}$) is of δ-*quality* if

$$\{\{Q\boldsymbol{\alpha}\}\} = \max_{1 \le i \le n} \left(\min_{p_i \in \mathbf{Z}} \left| Q \frac{a_i}{A} - p_i \right| \right) \tag{3.2}$$

satisfies

$$\{\{Q\boldsymbol{\alpha}\}\} \le A^{-\delta} . \tag{3.3}$$

By analogy with Proposition 2.1 we will say a denominator Q (or a vector $\boldsymbol{\xi}$) is an ‘‘unusually good’’ approximation if it gives a δ-quality approximation with $\delta > \frac{1}{n}$. This analogy is justified by the following result.

Theorem A. For $n \ge 2$ *the ensemble*

$$S_n^*(A) = \{ \boldsymbol{\alpha} = (\frac{a_1}{A}, \ldots, \frac{a_n}{A}) : \; 0 \le a_i < A \;\text{ with }\; g.c.d.\,(a_1, \ldots, a_n, A) = 1 \}$$

has at least $\frac{1}{2} A^n$ *members. Of these, at most* $O(A^{n(1-\delta)+1})$ *members have at least one* δ-*quality denominator.*

This theorem says that for any fixed $\delta > \frac{1}{n}$, as $A \to \infty$ at most an infinitesimal fraction of the members of this ensemble have at least one δ-quality approximation.

3. Shamir's attack on the basic Merkle-Hellman cryptosystem

Shamir's attack starts from the decryption congruence for the basic Merkle-Hellman cryptosystem, which is

$$W a_i \equiv s_i \pmod{M}; \quad 1 \le i \le n \;, \tag{3.1}$$

where the s_i are a superincreasing sequence, with s_1 the smallest member. We can rewrite (3.1) as

$$W a_i - M k_i = s_i \,, \quad 1 \le i \le n \tag{3.2}$$

where the $\{ k_i : \; 1 \le i \le n \}$ are nonnegative integers unknown to the cryptanalyst.

Shamir's attack has two main steps. The first step is that of finding the unknown k_i's in (3.2). Our object here is to show that this step is actually a simultaneous Diophantine approximation problem. To do this, we divide (3.2) by $M a_i$ to obtain

$$\frac{W}{M} - \frac{k_i}{a_i} = \frac{s_i}{M a_i} \;; \quad 1 \le i \le n \;. \tag{3.3}$$

For small values of i the right side of (3.3) is small, because the superincreasing property of the s_i implies that

$$s_i \le 2^{-n+i} M \;, \tag{3.4}$$

using the *size condition*

$$M > \sum_{i=1}^{n} s_i \;. \tag{3.5}$$

Also for ''almost all'' choices of the multiplier W in (3.1), we have

$$a_i \ge \frac{1}{n^2} M \;; \quad 1 \le i \le n \;. \tag{3.6}$$

Putting these estimates into (3.3) gives

$$\frac{W}{M} - \frac{k_i}{a_i} = O(2^{-n+i} n^2 M^{-1}) \tag{3.7}$$

This immediately gives

$$\frac{k_i}{a_i} - \frac{k_1}{a_1} = O(2^{-n+i} n^2 M^{-1}) \ , \tag{3.8}$$

and we expect that $k_i \approx a_i \approx M$ so that (3.8) implies

$$\frac{k_i}{k_1} - \frac{a_i}{a_1} \approx O(2^{-n+i} n^2 M^{-1}) \ . \tag{3.9}$$

Now equation (3.9) says that for any $d \geq 2$ the vector $\boldsymbol{\xi}_d = (\frac{k_2}{k_1}, \ldots, \frac{k_{d+1}}{k_1})$ is a reasonably good simultaneous Diophantine approximation to the vector $\boldsymbol{\alpha}_d = (\frac{a_2}{a_1}, \ldots, \frac{a_{d+1}}{a_1})$ constructed out of the public weights.

We can estimate how good a simultaneous Diophantine approximation $\boldsymbol{\xi}_d$ is to $\boldsymbol{\alpha}_d$ using the following heuristic argument. For a ‘‘random’’ denominator k_1 the expected size of the right side of (3.9) would be $O(M^{-1})$. Now (3.9) is actually smaller than this by a factor $\approx 2^{-n}$ when i is small, this factor arising from the fact that the beginning elements of the superincreasing sequence are small. We can relate this to the information rate R for the public weights by observing that

$$a_1 \approx M \approx 2^{nR^{-1}} \ .$$

Setting $D = R^{-1}$, equation (3.9) then gives approximately

$$\frac{k_i}{k_1} - \frac{a_i}{a_1} \approx O(2^{-n(1+D)}) = O(a_1^{-(1+R)}) \ . \tag{3.10}$$

It is possible to replace this heuristic argument by a rigorous one, to obtain the following result.

Theorem B. Let $\{s_i\colon\ 1 \leq i < n\}$ *be a superincreasing sequence with information rate R, and let* $\{a_i\colon\ 1 \leq i < n\}$ *be the corresponding public knapsack problem encrypted using* (W, M)*. Then for any M (satisfying the size condition) and ‘‘almost all’’ W, and for any* $d \geq 2$ *the vector*

$$\boldsymbol{\alpha}_d = \left[\frac{a_2}{a_1}, \ldots, \frac{a_{d+1}}{a_1} \right] \tag{3.11}$$

constructed from the public key has the δ*-quality approximation vector*

$$\boldsymbol{\xi}_d = \left[\frac{k_2}{k_1}, \ldots, \frac{k_{d+1}}{k_1} \right] \tag{3.12}$$

(with k_i*’s given by (3.2)) where*

$$\delta \geq R - \left[\frac{d + \log\ n}{Rn} \right] \ . \tag{3.13}$$

Theorem B shows that if we let the block length n of the cipher tend to infinity while holding the information rate R constant, then the d-dimensional vector $\boldsymbol{\alpha}_d$ has an $\approx R$-

quality simultaneous Diophantine approximation.

The heuristic underlying this first step in Shamir's algorithm is that if $\boldsymbol{\xi}_d$ is a ''sufficiently good'' approximation, then we might hope it is *uniquely* determined by (3.10). And if this is so, we might hope to find $\boldsymbol{\xi}_d$ using an algorithm to find good simultaneous Diophantine approximations. A *necessary* condition for $\boldsymbol{\xi}_d$ to be essentially uniquely determined by (3.10) is that $\boldsymbol{\xi}_d$ be an ''unusually good'' approximation to $\boldsymbol{\alpha}_d$, as defined in Section 2. Using this definition, Theorem B gives the following corollary.

Corollary C. For ''almost all'' basic knapsack cryptosystems with information rate R and sufficiently large block size n, the vector $\boldsymbol{\alpha}_d$ has at least one ''unusually good'' approximation $\boldsymbol{\xi}_d$ provided $d > R^{-1}$.

We can make ''sufficiently large'' in Corollary C quantitative using (3.13). In fact, being ''unusually good'' in a certain sense makes $\boldsymbol{\xi}_d$ ''almost uniquely'' determined by (3.10). Shamir [16] analyzes a model showing this true for information rates R with $1/2 < R \le 1$, and the analysis in [9] shows it holds for the full range $0 < R \le 1$. To rephrase this, the condition that $\boldsymbol{\xi}_d$ being ''unusually good'' is (almost always) a necessary and sufficient condition for it to be ''almost uniquely'' recoverable from (3.10).

Shamir [6] proposes using (3.10) to find the integer k_1 in (3.2) by solving the integer programming problem

$$|k_i a_1 - k_1 a_i| \le 2^{-n+d} n^2 M; \quad 1 \le i \le d \ , \tag{3.14}$$

for the unknown $(k_1, \ldots, k_d)$. The key role of Corollary C is to assert that the integer program can be taken to have the *fixed* number of variables $d = [R^{-1}] + 1$ when the information rate R is held fixed and the block size $n \to \infty$. Shamir [6] uses a polynomial-time algorithm of H. W. Lenstra, Jr. [13] for solving integer programs having a fixed number of variables to solve (3.14). Corollary C gives the dependence of the problem size on the information rate R. In particular the number of variables in (3.14) goes up as R decreases.

The second step of Shamir's attack is based on the observation that the transformation mapping superincreasing sequences $\{s_i: \ 1 \le i \le n\}$ and multiplier-modulus pairs (W, M) to public weights $\{a_i: \ 1 \le i \le n\}$ defined by

$$W a_i \equiv s_i \pmod{M}; \quad 1 \le i \le n \ ,$$

with $0 \le a_i < M$, is *not one-to-one*. In particular, any pair (W^*, M^*) with $\frac{W^*}{M^*}$ close enough to $\frac{W}{M}$ can function as a decryption multiplier-modulus pair. To see this, write $W^* = \lambda W$, $M^* = \lambda M + \varepsilon$ where ε is a small error and λ is a scaling factor. Substituting into (3.2), we obtain

$$\begin{aligned} W^* a_i - M^* k_i &= \lambda (W a_i - M k_i) + \varepsilon k_i \\ &= \lambda s_i + \varepsilon k_i \ . \end{aligned} \tag{3.15}$$

The right side of (3.15) is just the superincreasing sequence λs_i perturbed by the quantities

εk_i, so it will be superincreasing whenever ε is small enough. Now note that the first step of Shamir's algorithm found k_1, and (3.7) gives

$$\frac{W}{M} - \frac{k_1}{a_1} = O(2^{-n} m^2 M^{-1}) ,$$

i.e. the known quantity $\frac{k_1}{a_1}$ is a very good approximation to $\frac{W}{M}$. Shamir's observation for his second step was that an appropriate search of an interval near $\frac{k_1}{a_1}$ will nearly always produce a fraction $\frac{W^*}{M^*}$ for which the right side of (3.15) is superincreasing. Then (W^*, M^*) functions as a decryption trapdoor permitting the cryptanalyst to break the basic Merkle-Hellman cryptosystem.

4. Multiply Iterated Knapsacks

We now show there is an analogue of Theorem B for multiply iterated knapsacks. For simplicity we sketch the essential ideas for the case of doubly iterated knapsacks.

Let $\{a_i :\ 1 \le i \le n\}$ denote the public weights for a doubly iterated knapsack cryptosystem. The equations for decryption are

$$W_2 a_i - M_2 k_{2,i} = a_{1,i} ; \quad 1 \le i \le n , \tag{4.1}$$

$$W_1 a_{1,i} - M_1 k_{1,i} = s_i ; \quad 1 \le i \le n , \tag{4.2}$$

where

(i) $\{s_i :\ 1 \le i \le n\}$ is a superincreasing sequence.

(ii) We have $0 \le a_{1,i} < M_2$ and $0 \le s_i < M_1$ for $1 \le i \le n$.

(iii) The moduli M_2 and M_1 satisfy the *size conditions*

$$M_2 > \sum_{i=1}^{n} a_{1,i} ,$$

and

$$M_1 > \sum_{i=1}^{n} s_i .$$

We proceed by substituting the value for $a_{1,i}$ in (4.1) into (4.2) to obtain

$$W_2 W_1 a_i - M_2 W_1 k_{2,i} - M_1 k_{1,i} = s_i ; \quad 1 \le i \le n . \tag{4.3}$$

Dividing this equation by $M_2 W_1\, a_{2,i}$ gives

$$\frac{W_2}{M_2} - \frac{1}{a_i} \left[k_{2,i} + k_{1,i} \frac{M_1}{M_2 W_1} \right] = \frac{s_i}{M_2 W_1 a_{2,i}} . \tag{4.4}$$

In particular, the right side of (4.4) is ''unusually small'' for small values of i, since $\{s_i\}$ is superincreasing. To quantify this, note that if R is the information rate for the public weights $\{a_i\}$, and setting $D = R^{-1}$ we have

$$s_i \approx O(2^{(D-1)n+i}) \tag{4.5}$$

and, to a first approximation,

$$W_1 \approx M_1 \approx W_2 \approx M_2 \approx k_{1,i} \approx k_{2,i} \approx a_{1,i} \approx a_i \approx 2^{Dn} \tag{4.6}$$

holds for ''almost all'' choices of (W_1, M_1) and (W_2, M_2). Substituting (4.5) and (4.6) in (4.4) yields

$$\frac{W_2}{M_2} - \frac{1}{a_i} (k_{2,i} + k_{1,i} \frac{M_1}{M_2 W_1}) = O(2^{-(2D+1)n}) \tag{4.7}$$

for small i. Now by Dirichlet's theorem on Diophantine approximation we can always find a good approximation $\frac{r_1}{r_2}$ to $\frac{M_1}{M_2 W_1}$ such that

$$\left| \frac{M_1}{M_2 W_1} - \frac{r_1}{r_2} \right| \le 2^{-(2D+1)n} \tag{4.8}$$

where

$$1 \le r_2 \le 2^{-(\frac{2D+1}{2})n} . \tag{4.9}$$

Substituting $\frac{r_1}{r_2}$ for $\frac{M_1}{M_2 W_1}$ in (4.7) gives

$$\frac{W_2}{M_2} - \frac{1}{a_i r_2} (k_{2,i} r_2 + k_{1,i} r_1) = O(2^{-(2D+1)n}) , \tag{4.10}$$

using the estimates (4.6) and (4.8). Then subtracting equation (4.10) for $i = 1$ and $i = j$ gives

$$\frac{1}{a_1 r_2} (k_{2,1} r_2 + k_{1,1} r_1) - \frac{1}{a_j r_2} (k_{2,j} r_2 + k_{1,j} r_1) = O(2^{-(2D+1)n}) . \tag{4.11}$$

Multiplying this equation by $\frac{a_j r_2}{k_{2,1} r_2 + k_{1,1} r_1}$ gives

$$\frac{a_j}{a_1} - \frac{k_{2,j} r_2 + k_{1,j} r_1}{k_{2,1} r_2 + k_{1,1} r_1} \approx O(2^{-(2D+1)n}) ; \tag{4.12}$$

noting here that

$$Q = k_{2,1} r_2 + k_{1,1} r_1 = O(2^{(2D + \frac{1}{2})n}) ,$$

using (4.6) and (4.9). Then equation (4.12) says that, for any fixed d, there is a fairly good simultaneous Diophantine approximation to $\boldsymbol{\alpha}_d = (\frac{a_2}{a_1}, \ldots, \frac{a_d}{a_1})$ with denominator $Q = k_{2,1} r_2 + k_{1,1} r_1$. However the estimate for Q shows that in general Q is much larger than a_1. So we define

$$\hat{k}_1 \equiv k_{2,1} r_2 + k_{1,1} r_1 \pmod{a_1} \qquad (4.13)$$

with $0 \le \hat{k}_1 < a_1$, and we define t by

$$\hat{k}_1 = k_{2,1} r_2 + k_{1,1} r_1 - t a_1 \ . \qquad (4.14)$$

Next we *define* $\hat{k}_i$ for $2 \le i \le d+1$ by

$$\hat{k}_i = k_{2,i} r_2 + k_{1,i} r_1 - t a_i \ , \qquad (4.15)$$

where t was determined by (4.14). Then, *provided* $\hat{k}_1 \neq 0$, we obtain using (4.12), (4.14) and (4.15) that

$$\left| \frac{a_j}{a_1} - \frac{\hat{k}_j}{\hat{k}_1} \right| \approx O(2^{-(D + \frac{1}{2})n}) ; \quad 2 \le j \le d+1 \ . \qquad (4.16)$$

In particular, for sufficiently large d, this gives an ‘‘unusually good’’ simultaneous Diophantine approximation to $\boldsymbol{\alpha}_d = (\frac{a_2}{a_1}, \ldots, \frac{a_{d+1}}{a_1})$. The existence of the simultaneous Diophantine approximations $\boldsymbol{\xi}_d = (\frac{\hat{k}_2}{\hat{k}_1}, \ldots, \frac{\hat{k}_{d+1}}{\hat{k}_1})$ in (4.16) is justified by the following key lemma.

Lemma. *‘‘Almost always’’* $k_1 \neq 0$.

Proof. We argue by contradiction. Suppose $\hat{k}_1 = 0$, i.e.

$$k_{2,1} r_2 + k_{1,1} r_1 \equiv 0 \pmod{a_1} \ .$$

Then (4.10) gives

$$\frac{W_1}{M_1} - \frac{\mu_2}{r_2} = O(2^{-(2D+1)n}) \ , \qquad (4.17)$$

where

$$\mu_2 = \frac{1}{a_1} (k_{2,1} r_2 + k_{1,1} r_1) \qquad (4.18)$$

is now an integer. But if $\frac{W_2}{M_2} \neq \frac{\mu_2}{r_2}$ then

$$\left| \frac{W_2}{M_2} - \frac{\mu_2}{r_2} \right| \ge \frac{1}{M_2 r_2} \approx 2^{-(2D + \frac{1}{2})n} \ . \qquad (4.19)$$

This contradicts (4.17) so that in fact $\frac{W_2}{M_2} = \frac{\mu_2}{r_2}$ must hold.

Since $(W_2, M_2) = 1$, we must have

$$r_2 = M_2 k$$

$$\mu_2 = W_2 k$$

where $k = O(2^{\frac{1}{2} n})$ using (4.9). Substituting this in (4.8) gives

$$\left| \frac{M_1}{M_2 W_1} - \frac{r_1}{M_2 k} \right| < 2^{-(2D+1)n} ,$$

hence multiplying by M_2 gives

$$\left| \frac{M_1}{W_1} - \frac{r_1}{k} \right| < 2^{-(D+1)n} . \tag{4.20}$$

But now if $\frac{M_1}{W_1} \neq \frac{r_1}{k}$ then

$$\left| \frac{M_1}{W_1} - \frac{r_1}{k} \right| \geq \frac{1}{W_1 k} \approx 2^{-(D + \frac{1}{2})n} . \tag{4.21}$$

This now contradicts (4.20), so we must have

$$\frac{M_1}{W_1} = \frac{r_1}{k} .$$

Now we have a final contradiction, because $(M_1, W_1) = 1$ is in lowest terms, while ‘‘almost always’’ $W_1 \approx 2^{Dn}$ and the denominator $k = O(2^{\frac{1}{2} n})$ is smaller than W_1 since $D \geq 1$. ■

This completes the sketch of the argument for doubly iterated knapsacks. The extension to *M*-times iterated knapsacks is based on the equation

$$\frac{W_m}{M_m} - \frac{1}{a_i} \left[k_{m,\,i} + k_{m-1,\,i} \frac{M_{m-1}}{M_m W_{m-1}} + k_{m-2,\,i} \frac{M_{m-2}}{M_m W_{m-1} W_{m-2}} + \ldots + \right.$$

$$\left. k_{1,\,i} \frac{M_1}{M_m W_{m-1} W_{m-2} \ldots W_1} \right] = O(2^{-(mD+1)n}) \tag{4.22}$$

analogous to (4.7). In this case we use Dirichlet’s theorem for simultaneous Diophantine approximation to assert the existence of a vector $(\frac{r_{m-1}}{r_m}, \ldots, \frac{r_1}{r_m})$ which approximates $(\frac{M_{m-1}}{M_m W_{m-1}}, \frac{M_{m-2}}{M_m W_{m-1} W_{m-2}}, \ldots, \frac{M_1}{M_m W_{m-1} \ldots W_1})$ so that

$$\left| \frac{M_{m-1}}{M_m W_{m-1} \ldots W_j} - \frac{r_j}{r_m} \right| \leq 2^{-(mD+1)n}; \quad 1 \leq j \leq m , \tag{4.23}$$

with

$$1 \le r_m \le 2^{(mD+1)n(1 - \frac{1}{m})} . \tag{4.24}$$

The ‘‘unusually good’’ simultaneous Diophantine approximation $\xi_d = (\frac{\hat{k}_2}{\hat{k}_1}, \ldots, \frac{\hat{k}_d}{\hat{k}_1})$ to $\boldsymbol{\alpha}_d$ then has

$$\hat{k}_1 = k_{m,1} r_m + k_{m-1,1} r_{m-1} + \ldots + k_{1,1} r_1 - t a_1 \tag{4.25}$$

with $0 \le \hat{k}_1 < a_1$. The analogue of the lemma holds: ‘‘almost always’’ $\hat{k}_1 \ne 0$.

We obtain the following quantitative result.

Theorem D. Let $\{s_i : 1 \le i \le n\}$ *be a superincreasing sequence with information rate* R*, and let* $\{a_i : 1 \le i \le n\}$ *be the corresponding* M*-times iterated public knapsack problem with decryption multiplier-modulus pairs* $(W_m, M_m), \ldots, (W_1, M_1)$ *satisfying the size conditions. Then for ‘‘almost all’’ such pairs and for* $d \ge 2$ *the vector*

$$\boldsymbol{\alpha}_d = \left[\frac{a_2}{a_1}, \ldots, \frac{a_{d+1}}{a_1} \right]$$

constructed from the public keys has the δ*-quality approximation vector*

$$\xi_d = \left[\frac{\hat{k}_2}{\hat{k}_1}, \ldots, \frac{\hat{k}_{d+1}}{\hat{k}_1} \right] \tag{4.26}$$

(with $\hat{k}_1$ *given by* (4.14)*)) where*

$$\delta > \frac{1}{m} R - m \left[\frac{d + \log n}{Rn} \right] . \tag{4.27}$$

We remark that since there are generally several solutions to (4.8), there are not one but many such vectors ξ_d in (4.25). (See Conjecture F in Section 5.)

Using the notion of ‘‘unusually good’’ simultaneous Diophantine approximation given in Section 2, we obtain the following corollary.

Corollary E. For ‘‘almost all’’ m*-times iterated knapsack cryptosystems with information rate* R *and sufficiently large block size* n*, the vector* $\boldsymbol{\alpha}_d$ *has at least one ‘‘unusually good’’ approximation* ξ_d *provided* $d > mR^{-1}$.

We can quantify ‘‘sufficiently large’’ using (4.27). As in the basic knapsack case, such ‘‘unusually good’’ approximations can be recovered either by using the integer program in a fixed number of variables

$$| \hat{k}_i a_1 - \hat{k}_1 a_i | \le 2^{-\frac{n}{m}} \tilde{M}; \quad 2 \le i \le d+1 ,$$

analogous to (3.14), or by using a lattice basis reduction algorithm as described in the next section.

In order to use Theorem D as the basis of an attack on iterated knapsacks, we need to know that the vector $\boldsymbol{\alpha}_d$ does not have many extraneous ''unusually good'' simultaneous Diophantine approximations other than the vectors $\boldsymbol{\xi}_d$ that are determined by (4.14), (4.15) for some $\frac{r_1}{r_2}$ satisfying (4.8). Empirical tests indicate this to be the case. It is possible that the arguments of [9], [10] can be extended to prove this rigorously; however I have not done this.

5. Short Vector in Lattice Problems

In order to exploit the existence of ''unusually good'' simultaneous Diophantine approximations, we need an algorithm to locate them. Shamir's attack on the basic Merkle-Hellman scheme used the fact that for a fixed information rate R, the simultaneous Diophantine approximation problem involved is an integer programming problem in a fixed number of variables, where the number of variables depends on R as given in Corollary C. He then used a polynomial-time algorithm of H. W. Lenstra, Jr. [13] for solving integer programs in a fixed number of variables. However this algorithm is impractical for large numbers of variables. Len Adleman [2] observed that the recently developed L^3 algorithm for finding a reduced basis of an integral lattice (due to A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovasz [12]) provides a much more efficient way of finding the desired ''unusually good'' simultaneous Diophantine approximations.

The simultaneous Diophantine approximation problem is easily set up as a problem of finding a short vector in a lattice L. Suppose we are looking for a denominator Q that produces at least a δ-quality approximation vector $\boldsymbol{\xi}_d = (\frac{p_1}{Q}, \ldots, \frac{p_d}{Q})$ to $\boldsymbol{\alpha}_d = (\frac{a_2}{a_1}, \ldots, \frac{a_{d+1}}{a_1})$. We consider the lattice $L \subseteq \mathbf{Z}^{d+1}$ with basis vectors

$$
\begin{aligned}
\mathbf{b}_1 &= (\lambda a_1, 0, \ldots, 0, 0) , \\
\mathbf{b}_2 &= (0, \lambda a_1, \ldots, 0, 0) , \\
&\vdots \\
\mathbf{b}_d &= (0, 0, \ldots, \lambda a_1, 0) , \\
\mathbf{b}_{d+1} &= (-\lambda a_2, -\lambda a_3, \ldots, -\lambda a_{d+1}, 1) .
\end{aligned}
$$

where λ is a scaling factor, chosen to be an integer of approximate size

$$\lambda \approx (a_1)^{\delta} . \tag{5.2}$$

If $\boldsymbol{\alpha}$ has an approximation $\boldsymbol{\xi}$ of δ-quality, then the vector

$$
\begin{aligned}
\mathbf{w} &= p_1 \mathbf{b}_1 + \ldots + p_d \mathbf{b}_d + Q \mathbf{b}_{d+1} \\
&= (\lambda(p_1 a_1 - Q a_2), \lambda(p_2 a_1 - Q a_3), \ldots, \lambda(p_n a_1 - Q a_{d+1}), Q)
\end{aligned} \tag{5.3}
$$

in the lattice L has Euclidean length

$$\|\mathbf{w}\| \lesssim n^{1/2}\ a_1\ , \tag{5.4}$$

which is quite short compared to the original basis vectors, which are of length on the order of $(a_1)^{1+\delta}$. In running the L^3 algorithm, we hope that the vector **w** will show up in the reduced basis for the lattice obtained by the algorithm. Note that the desired denominator Q of ξ is thus directly recoverable as the last coordinate of **w**. Also note that there is always another short vector

$$\mathbf{x} = (0,\ 0,\ldots,\ 0,\ a_1) = a_2\mathbf{b}_1 + a_3\mathbf{b}_2 + \ldots + a_{d+1}\mathbf{b}_d + a_1\mathbf{b}_{d+1}\ , \tag{5.5}$$

in L, and that **x** is always linearly independent of **w**. The following conjecture seems to describe what happens for lattices generated by iterated knapsacks. Call the lattice L given by (5.1) $L(a_1,\ldots,\ a_{d+1};\ \delta)$.

Conjecture F. For ''almost all'' lattices $L(a_1,\ldots,\ a_{d+1};\ \delta)$ *arising from an* m*-times iterated knapsack with information rate* R *and where*

$$\delta \approx \frac{1}{m}\ R\ , \tag{5.6}$$

$$d = \beta\ mR^{-1}\ , \tag{5.7}$$

with $\beta \geq 1$*, the reduced basis produced by the* L^3 *algorithm contains exactly* $m+1$ *''short'' vectors of length on the order of* a_1*, one of which is* **x** *and the other* m *are each associated to a vector* ξ_d *given by Theorem* D*. The remaining* $d-m$ *''long'' vectors in the reduced basis are of length on the order of* $(a_1)^{1+\delta(1-\frac{1-R}{\beta-R})}$.

Note that by increasing d, i.e. increasing β, we can increase the separation in size between the $m+1$ ''short'' vectors and the $d-m$ ''long'' vectors. This heuristic has been verified experimentally on examples of doubly and triply iterated knapsacks by E. F. Bickell and A. M. Odlyzko.

I believe it should be possible to prove that for ''almost all'' lattices $L(a_1,\ldots,\ a_d,\ \delta)$ arising from m-times iterated knapsacks there exists an $m+1$-dimensional subspace of $L(a_1,\ldots,\ a_{d+1},\ \delta)$ that is spanned by the vectors **x** and m *different* ξ_d arising from Theorem D, which are all of length $O(a_1)$. The problem of proving a rigorous version of Conjecture F may be difficult.

6. A Non-linear Diophantine Approximation Problem

We next consider the problem of recovering the information contained in the ''unusually good'' simultaneous Diophantine approximations $\xi_d = (\frac{\hat{k}_2}{\hat{k}_1},\ldots,\ \frac{\hat{k}_{d+1}}{\hat{k}_1})$ to $\boldsymbol{\alpha}_d = (\frac{a_2}{a_1},\ldots,\ \frac{a_{d+1}}{a_1})$ guaranteed to exist for multiply iterated knapsacks by Theorem D.

Suppose we are given an m-times iterated knapsack and that we have successfully found $(\hat{k}_1,\ldots,\ \hat{k}_{d+1})$ where

$$\hat{k}_i = k_{m,\,i} r_m + \ldots + k_{1,\,i} r_1 - t a_i ; \quad 1 \le i \le d+1 \ , \tag{6.1}$$

a set of equations in which only the $\hat{k}_i$ and a_i are known. Can we recover the unknowns $\{k_{j,\,i} : \ 1 \le i \le d+1, \ 1 \le j \le m\}$ and $\{r_i : \ 1 \le i \le d\}$ and t from (6.1)? This is a non-linear problem because it involves the terms $k_{j,\,i} r_j$ which are bilinear in the unknowns. It may appear that this system is underdetermined and that there is not enough information in (6.1) to recover all these unknowns. We advance a heuristic argument to the effect that (6.1) does determine all these unknowns, *provided there are appropriate size restrictions on the variables and* d *is large enough.* For simplicity we discuss the doubly-iterated case where (6.1) becomes

$$\hat{k}_i = k_{2,\,i} r_2 + k_{1,\,i} r_1 - t a_i ; \quad 1 \le i \le d+1 \ . \tag{6.2}$$

We assume that the doubly-iterated knapsack transmits information at rate R, and we set $D = R^{-1}$. The *size restrictions* we use are as follows.

(i) r_1 and r_2 are not too small. More precisely, using (4.6) and (4.8), we have

$$r_1 \approx 2^{\frac{1}{2} n} \ ,$$

$$r_2 \approx 2^{(D + \frac{1}{2}) n} \ .$$

(ii) The $k_{1,\,i}$ and $k_{2,\,i}$ are nonnegative and are not too large. More precisely, using (4.6),

$$k_{1,\,i} \approx 2^{Dn}$$

$$k_{2,\,i} \approx 2^{Dn} \ .$$

Now we can state the heuristic argument.

Heuristic Lemma. Under the size restrictions r_1 *and* r_2 *ought to be uniquely determined by (6.2) when* $d \ge 4(R^{-1} + 1)$.

Heuristic ''proof''. Consider (r_1, r_2) as fixed, with r_1 drawn from the interval $[1, 2^{\frac{1}{2} n}]$ and r_2 from the interval $[1, 2^{(D + \frac{1}{2}) n}]$, so there are $2^{(D+1)n}$ such pairs. For each such pair, there are exactly 2^{2Dn} numbers of the form

$$\Rightarrow \ = x_1 r_1 + x_2 r_2 \quad ; \ x_1, \ x_2 \in [1, 2^{Dn}] \tag{6.3}$$

and all such integers fall in $[1, 2^{(2D + \frac{1}{2}) n + 1}]$. Hence the probability that a given integer $\Rightarrow^* \in [1, 2^{(2D + \frac{1}{2}) n + 1}]$ is of the form (6.3) is approximately $2^{-\frac{1}{2} n}$. Assuming the heuristic that the sets

$$S(r_1, r_2) = \{x_1 r_1 + x_2 r_2 : \ x_1, x_2 \in [1, 2^{Dn}]\} \tag{6.4}$$

are ''approximately independent'' for different pairs (r_1, r_2), $2(D+1)$ random draws of integers of the form (6.3) ought to determine (r_1, r_2) uniquely since there are $2^{(D+1)n}$ such pairs and each draw of an element $\Rightarrow$ in (6.3) should cut down the admissible pairs (r_1, r_2)

by a factor $2^{-\frac{1}{2}n}$.

Now we consider (6.2) which has the additional complication of the unknown *t*, where $t \approx 2^{(\frac{D+1}{2})n}$. Now (6.2) can be rewritten:

$$\hat{k}_i + ta_i = \rightarrowtail_i \;\; ; \;\; \rightarrowtail_i \in S(r_1, r_2) \; .$$

Again assuming a similar heuristic, we need an extra $2(\frac{D+1}{2})$ draws of $\rightarrowtail_i$ to eliminate the ambiguity of the variable *t*, i.e. we expect that when $d \geq 4(D+1)$ then on average at most one of the $2^{(D+1)n}$ sets

$$T(u) = \{\hat{k}_i + ua_i : \; 1 \leq i \leq d\} \; ; \quad 1 \leq u \leq 2^{(D+\frac{1}{2})n} \; ,$$

will come from *any* $S(r_1, r_2)$ and that one set $T(t)$ will come from *exactly one* $S(r_1, r_2)$, which is the one we want. This incidentally determines *t*. ■

Next we note that the unknown *t* can be eliminated from the system (6.1). We have

$$\hat{k}_i a_1 - \hat{k}_1 a_i = (k_{m,\,i} a_1 - k_{m,\,1} a_i) r_m + \ldots + (k_{1,\,i} a_1 - k_{1,\,1} a_i) r_1 \; ; \quad 2 \leq i \leq d+1 \; , \tag{6.5}$$

which is a system of the form

$$\hat{k}_i^* = \rightarrowtail_{m,\,i} r_m + \ldots + \rightarrowtail_{m,\,1} r_1 \; ; \quad 1 \leq i \leq d \; .$$

Hence we are led to consider the following general problem.

Problem G. Given a test set $T = \{\hat{k}_i : \; 1 \leq i \leq d\}$. *Find a vector* $(r_1, \ldots, r_m)$ *with* $r_i \in [1, 2^{\alpha n}]$ *such that*

$$T \subseteq S(r_1, \ldots, r_m)$$

where

$$S(r_1, \ldots, r_m) = \{x_1 r_1 + \ldots + x_m r_m : \text{ all } \; x_i \in [-2^{\beta n}, 2^{\beta n}]\} \; ,$$

provided $\alpha > (m-1)\beta$, *and*

$$d \geq \frac{\alpha + \beta}{\alpha - (m-1)\beta} \; , \tag{6.6}$$

when one exists.

The bound (6.6) for *d* is that suggested by the heuristic argument in order that $(r_1, \ldots, r_m)$ be essentially uniquely determined by *T*.

Problem G has an easy solution for $m = 1$, since in that case $T = \{\hat{k}_i : \; 1 \leq i \leq d\}$ where

$$\hat{k}_i = \lambda_i r_1$$

and so we can generally take

$$r_1 = g.c.d.\; (\hat{k}_1, \ldots, \hat{k}_d) \; . \tag{6.7}$$

I do not know how to solve Problem G in polynomial time (in n) for any $m \geq 2$.

7. An attack on doubly iterated knapsacks

We now describe an *ad hoc* attack for breaking doubly iterated knapsack cryptosystems. It is based on the fact that we can, by a trick, reduce it to a case of Problem G for $m = 1$, which has the easy solution (6.7).

We suppose that the doubly iterated knapsack has information rate R, and we set $D = R^{-1}$ and choose

$$d = 2\beta R^{-1} \tag{7.1}$$

for some *fixed* $\beta > 1$.

Step 1. Use the L^3 algorithm on the $d+1$ dimensional lattice $L(a_1, \ldots, a_{d+1};\ \frac{1}{2} R)$ described in Section 5 to find $(\hat{k}_1, \ldots, \hat{k}_{d+1})$ such that

$$\hat{k}_i = k_{2,\,i} r_2 + k_{1,\,i} r_1 - t a_i \ ; \quad 1 \leq i \leq d+1 \ . \tag{7.2}$$

(The bound (7.1) for d is necessary for step 1 to work.)

Step 2. Use the L^3 algorithm on the $n+1$ dimensional lattice $L(a_1, \ldots, a_n\ ;\ \frac{\log n}{n})$ to recover $(k_{2,\,1}, \ldots, k_{2,\,n})$.

Step 3. Using the fact that $\hat{k}_i$, $k_{2,\,i}$ and a_i are now known in (7.2), use them to eliminate r_2 and t from (7.2). One way to do this is to first eliminate r_2, obtaining

$$\hat{k}_i k_{2,\,1} - \hat{k}_1 k_{2,\,i} = (k_{1,\,i} k_{2,\,1} - k_{1,\,1} k_{2,\,i}) r_1 - t(k_{2,\,1} a_i - k_{2,\,i} a_1) \ ; \quad 2 \leq i \leq d+1 \ . \tag{7.3}$$

We rewrite (7.3) as

$$\hat{\ell}_i = m_i r_1 + t \hat{n}_i \ ; \quad 2 \leq i \leq d+1 \ , \tag{7.4}$$

where $\hat{\ell}_i$ and $\hat{n}_i$ are known. Then eliminate t from (7.4) obtaining

$$\hat{\ell}_i \hat{n}_2 - \hat{\ell}_2 \hat{n}_i = (m_i \hat{n}_2 - m_2 \hat{n}_i) r_1 \ ; \quad 3 \leq i \leq d+1 \ . \tag{7.5}$$

Now the left side of (7.5) is known, so a multiple of r_1 is found by a *g.c.d.* calculation (as in (6.7)). A second way to proceed is to eliminate first t and then r_2; this leads to a different set of equations of the same form as (7.5). Then take the *g.c.d.*s of this second set of equations, to determine a second multiple of r_1. The *g.c.d.* of these two multiples of r_1 is generally r_1.

Step 4. Now that r_1 is known, the equation (7.2) is linearized. We can now use the L^3 algorithm on the $d+4$ dimensional lattice L^* with basis

$$\mathbf{b}_1 = (r_1,\ 0, \ldots,\ 0,\ 0,\ 0,\ 0)\ ,$$

$$\mathbf{b}_2 = (0,\ r_1,\ \ldots,\ 0,\ 0,\ 0,\ 0)\ ,$$

$$\ldots$$

$$\mathbf{b}_{d+1} = (0,\ 0,\ \ldots,\ r_1,\ 0,\ 0,\ 0)\ ,$$

$$\mathbf{b}_{d+2} = (\hat{k}_1, \ldots,\ \hat{k}_{d+1},\ 1,\ 0,\ 0)\ ,$$

$$\mathbf{b}_{d+3} = (k_{2,\,1}, \ldots,\ k_{2,\,d+1},\ 0,\ 2^{-(D+\frac{1}{2})n},\ 0)\ ,$$

$$\mathbf{b}_{d+4} = (a_1, \ldots,\ a_{d+1},\ 0,\ 0,\ 2^{-(D+\frac{1}{2})n})\ .$$

The short vector **w** in the lattice that we are looking for is given by

$$\mathbf{w} = \sum_{i=1}^{d+1} k_{1,\,i}\mathbf{b}_i - \mathbf{b}_{d+2} + r_2\mathbf{b}_{d+3} - t\mathbf{b}_{d+4}$$

$$= (0,\ 0, \ldots,\ 0,\ -1,\ r_2 2^{-(D+\frac{1}{2})n},\ -t2^{-(D+\frac{1}{2})n})\ .$$

(Actually the basis vectors in L^* are scaled up by a factor $2^{(D+\frac{1}{2})n}$ so as to have integral components.) We expect to recover in this way r_2, t and all the $k_{1,\,i}$.

Step 5. We have from (4.7) that

$$\frac{W_2}{M_2} - \frac{1}{a_1 r_1}(k_{2,\,1}r_2 + k_{1,\,1}r_1) = O(2^{-(2D+1)n})\ . \tag{7.7}$$

Then $\frac{W_2}{M_2}$ will be a convergent in the continued fraction expansion of $\frac{k_{2,\,1}r_2 + k_{1,\,1}r_1}{a_1 r_1}$ which is followed by a huge partial quotient. This determines W_2 and M_2 uniquely.

Step 6. Use $(W_2,\ M_2)$ to strip away the outer modular multiplication. We obtain a basic Merkle-Hellman knapsack which can be broken by Shamir's algorithm. Alternatively, note that from (4.8) we have

$$\left| \frac{M_1}{W_1} - \frac{r_1 M_2}{r_2} \right| < 2^{-(D+1)n}\ .$$

Hence $(r_2,\ r_1 M_2)$ will usually perform well enough to be the ‘‘approximate’’ decrypt pair $(W_1^*,\ M_1^*)$ in Shamir's attack. ■

The trick used in this attack was step 2, which is essentially the first step in Adleman's attack [1,2] on multiply iterated knapsacks. This step is the slowest and also seems to be the most problematical in the attack described above, see [4].

Empirical testing of this algorithm indicates that Steps 1, 3 and 5 all perform well. In Step 3 it appears necessary to do both eliminations and then take a *g.c.d.*, since each elimination run separately seems to locate only a huge multiple of r_1. Step 4 has not been tested yet. Finally note that if the public weights $\{a_i : 1 \le i \le n\}$ are a *permuted* version of the superincreasing weights, then step 1 has to be run on $n\binom{n}{d}$ choices of $d+1$ of the public weights, to locate a_1 and the correct set $\{a_2, a_3, \ldots, a_{d+1}\}$. Since d is fixed by (7.1), this takes polynomial time in n as $n \to \infty$ with R fixed.

References

[1] L. Adleman, On Breaking the Iterated Merkle-Hellman Public Key Cryptosystem, in: Advances in Cryptology, Proceedings of Crypto-82 (Eds: D. Chaum, R. Rivest, A. T. Sherman), Plenum Press 1983, 303-308.

[2] L. Adleman, On Breaking Generalized Knapsack Public Key Cryptosytems, Proc. 15th Annual ACM Symposium on Theory of Computing, 1983, 402-412.

[3] E. F. Brickell, Solving low-density knapsacks, these proceedings.

[4] E. F. Brickell, J. C. Lagarias and A. M. Odlyzko, Evaluation of Adleman's Attack on Multiply Iterated Knapsacks (Abstract), these proceedings.

[5] E. F. Brickell and G. J. Simmons, A Status Report on Knapsack Based Public Key Cryptosystems, Congressus Numerantium *37* (1983), 3-72.

[6] E. F. Brickell, J. A. Davis, and G. J. Simmons, A Preliminary Report on the Cryptanalysis of Merkle-Hellman Knapsack Cryptosystems, in: Advances in Cryptalogy, Proceedings of Crypto-82 (Eds: D. Chaum, R. Rivest, A. T. Sherman), Plenum Press, New York 1983, 289-301.

[7] Y. Desmedt, J. Vandewalle, R. Govaerts, A Critical Analysis of the Security of Knapsack Public Key Algorithms, preprint.

[8] J. C. Lagarias, The Computational Complexity of Simultaneous Diophantine Approximation Problems, Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science (1982), 32-39. (to appear: SIAM J. Computing.)

[9] J. C. Lagarias, Performance Analysis of Shamir's Attack on the Basic Merkle-Hellman Knapsack Public Key Cryptosystem, in preparation.

[10] J. C. Lagarias, Simultaneous Diophantine Approximation of Rationals by Rationals, preprint.

[11] J. C. Lagarias and A. M. Odlyzko, Solving Low-Density Subset Sum Problems, Proc. 24th Annual IEEE Symposium on Foundations of Computer Science (1983), 1-10.

[12] A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz, Factoring polynomials with rational coefficients, Math. Annalen. *261* (1982), 515-534.

[13] H. W. Lenstra, Jr., Integer programming with a fixed number of variables, Math. of Operations Research, to appear.

[14] R. Merkle and M. Hellman, Hiding Information and Signatures in Trapdoor Knapsacks, IEEE Trans. Information Theory IT-24 (1978), 525-530.

[15] A. M. Odlyzko, Cryptanalytic attacks on the multiplicative knapsack cryptosystem and on Shamir's fast signature scheme, IEEE Trans. Information Theory, to appear.

[16] A. Shamir, A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem, Proc. 23rd Annual Symposium on Foundations of Computer Science (1982), 145-152.