On Breaking Generalized Knapsack Public Key Cryptosystems

(abstract)

by

Leonard M. Adleman+

University of Southern California

and

Massachusetts Institute of Technology

I. INTRODUCTION

In 1976 Diffie and Hellman introduced the concept of a public-key cryptosystem [1]. In 1977 Rivest, Shamir and Adleman discovered the first incarnation of such a system [9], and soon afterwards Merkle and Hellman produced a second one [7]. Despite great interest in the area, the years have produced no other public-key cryptosystems which have attracted wide spread attention.

The Merkle-Hellman system is based on the knapsack problem, and in the original paper on the topic, both a basic system and an iterated one were presented. In April of 1982, Adi Shamir demonstrated that the basic system was insecure [8].

In this paper new methods, generalizing those of Shamir, are presented for attacking generalizations of the basic system. It is shown how these methods may be applied to the Graham-Shamir public-key cryptosystem [2], and the iterated Merkle-Hellman public-key cryptosystem. We are unable to present a rigorous proof that the attacks presented here are effective. However, in the case of the Graham-Shamir system, the methods have been implemented and have performed well in tests.

The method of attack uses recent results of Lenstra, Lenstra, and Lovasz [5]. The cryptanalytic problem is treated as a lattice problem rather than a linear programming one as in Shamir's result.

II. GRAHAM SHAMIR KNAPSACK(GSK) [2]

Public-key cryptosystems require the generation of a "mated pair" of keys. One key is kept secret, the other is made public. It is crucial that the problem of computing the secret key from the public

+Research sponsored by National Science Foundation, Grant #MCS-8022533

key be intractable. In many GSK's this is apparently not the case. Below is a description of the procedure used to generate a mated pair of keys for such a system. How these keys are used for encryption and decryption will not concern us.

STEP 0:

Generate positive integers z, n.
Generate a sequence $c_1, c_2, \ldots, c_n$
with $c_i \in \{0, 1\}^z$, $i = 1, 2, \ldots, n$
such that

$$c_i \geq \sum_{j=1}^{i-1} c_j \quad i = 1, 2, \ldots, n$$

(where appropriate we will treat strings as the numbers they represent in binary).
Such a sequence is said to be "super increasing".
Note that for large z, n and small i, c_i will have leading zeros.

STEP 1:

Generate a positive integer y.
Generate a sequence $r_{h,1}, r_{h,2}, \ldots, r_{h,n}, r_{\ell,1}, r_{\ell,2}, \ldots, r_{\ell,n}$ with $r_{h,i} \in \{0,1\}^y$, $r_{\ell,i} \in \{0, 1\}^y$ $i = 1, 2, \ldots, n$

Calculate

$b_i = r_{h,i} * c_i * r_{\ell,i}$ $i = 1, 2, \ldots, n$
(the idea is that random r's well obscure the "super increasing" properties of the c's).

STEP 2:

Generate positive integers w, m, such that
a) $(w, m) = 1$
b) $m > \sum_{i=1}^{n} b_i$
Calculate

$a_i \equiv wb_i \bmod(m)$ $i = 1, 2, \ldots, n$

STEP 3:

Generate a permutation π on $\{1,2,\ldots,n\}$
output as the public key $< a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)} >$
keep $<w, m, <a_{\pi(1)}, \ldots, a_{\pi(n)}>>$ as the private key.

II.b) HUERISTIC FOR CRYPTANALYZING GSK's

We wish to recover w, m. Clearly, it is enough to recover $\ell \equiv w^{-1} \bmod(m)$ and m. We know that there are natural numbers $k_1, \ldots, k_n$ such that

$$la_i - k_i m = b_i \qquad i = 1, 2, \ldots, n \quad .$$

To begin the attack, the cryptanalyst randomly chooses a d element subset T of the published a's. How large d should be will be analyzed in what follows.

$$\text{Let } S = \{i \mid a_i \in T\} \quad ,$$

the i's in S will not be known to the cryptanalyst.
Consider the following system of equations:

SI $\qquad La_i - K_iM = B_i \qquad i \in S.$

SI has the following properties:

PI (Good) Among the solutions $\langle L, M, \langle K_i \rangle_{i \in S}, \langle B_i \rangle_{i \in S} \rangle$ is the "desired" one $\langle \ell, m \langle k_i \rangle_{i \in S} \langle b_i \rangle_{i \in S} \rangle$.

P2 (Bad) There are infinitely many undesirable solutions.

P3 (Bad) The system is non-linear ($K_i M$ terms) and there are no known polynomial time algorithms to solve such systems in general. In fact, the problem of solving even single equations with two unknowns of degree two is already NP-complete [6].

Curiously, P2 will be the key to overcoming P3. But to begin, we will simplify SI. By construction $M > b_i$, $i = 1, 2, \ldots, n$. Therefore, there is a largest e such that $M/2^e > b_i$, $i \in S$. We will assume that this e is known to the cryptanalyst (since at worst all possible e's could be tried in parallel). The size of this e plays an important role in determining the prospects that this attack will succeed. The larger the e with respect to M, the greater the chance of success. We now consider SII.

SII $\qquad 0 < La_i - K_iM \le M/2^e \qquad i \in S.$

This system has properties similar to those of SI:

P4 (Good) Among the solution $\langle L, M, \langle K_i \rangle_{i \in S} \rangle$ is the "desired" one $\langle \ell, m, \langle k_i \rangle_{i \in S} \rangle$.

P5 (Bad) There are infinitely many undesirable solutions.

P6 (Bad) The system is non-linear.

Concerning P5, not only are there infinitely many solutions, but in fact for large enough integer f there is always a solution of the form $\langle L, f, \langle K_i \rangle_{i \in S} \rangle$.

To see this consider the rational number $\frac{f}{m}$, $f > 0$ and let $\left\lfloor \frac{f}{m} \right\rfloor$ denote the nearest integer smaller than $\frac{f}{m}$. Then for some ε, $0 \le \varepsilon < 1$ $\frac{f}{m} - \varepsilon = \left\lfloor \frac{f}{m} \right\rfloor$. We know that

$$0 < \ell a_i - k_i m < m/2^e \qquad i \in S$$

so multiplying by f/m we have

$$0 < \frac{f}{m} \ell a_i - k_i f \le f/2^e$$

and it follows that for

$$f > 2^e \ell a, \quad a = \max_{i \in S} \{a_i\} \quad ,$$

$$0 < \left[\frac{f}{m}\,\ell a_i - \varepsilon\ell a_i\right] - k_i f \leq f/2^e - \varepsilon\ell a_i, \quad i \in S$$

and therefore

$$(*) \qquad 0 < \left\lfloor\frac{f}{m}\right\rfloor \ell a_i - k_i\, f \leq f/2^e\ , \ i \in S\ .$$

Since $g = 2^e\ \mathrm{MAX}(a_i)$ is approximately m and $m > \ell$ the choice of $f > 2g^2$ should give a new system with new properties:

<u>SIII</u> $\qquad 0 < La_i - K_i f \leq f/2^e \quad i \in S.$

<u>P7 (Good)</u> SIII is linear. Therefore, the methods of Lenstra, Lenstra and Lovacz, may be applicable in finding solutions.

<u>P8 (Very Good)</u>

a) Among the solutions of SIII there is at least one of the form $\langle L, \{k_i\}_{i \in S} \rangle$ where k_i's are exactly those which occur in the "desired" solution to SII. This follows immediately from (*) .

b) For $L = 1, 2, \ldots, \lfloor f/2^e a \rfloor$, there are solutions to SIII of the form $\langle L, \{0\}_{i \in S} \rangle$. This again follows immediately from (*).

c) For sufficiently large d there is a high probability that SIII has no other solutions than those indicated in a) and b).

By the arguments in [8], [3] we know that for d such that $2^{ed} > 2m$ system .

<u>SIV</u> $\qquad 0 < La_i - K_i m \leq m/2^e \quad i \in S$

should have no solution for L real, other than in intervals $(0, \varepsilon_1)$, (ℓ, ε_2) for some positive $\varepsilon_1, \varepsilon_2 < 1$.

Since there is a 1 - 1 correspondence between solution to SIII (for real L) and solutions to SIV, 8c follows.

Notice that Shamir's arguments also support the following: for integer h, $2 < h < 2^e$ and for d such that $h^d > 2m$ the system.

<u>SV</u> $\qquad 0 < La_i - K_i\, m \leq m/h \quad i \in S$

should have no solution for L real, other than in intervals $(0, \varepsilon_1)$, (ℓ, ε_2) for some positive $\varepsilon_1, \varepsilon_2 < 1$. And therefore

<u>SVI</u> $\qquad 0 < La_i - K_i f \leq f/h \quad i \in S$

should have no solutions in integers other than those of the form

a) $\langle L, \langle k_i\rangle_{i \in S} \rangle$

b) For $L = 1, 2, \ldots, \lfloor f/ha \rfloor$, $\langle L, \{0\}_{i \in S} \rangle$

In otherwords for sufficiently large d, other than the L's in solutions a) and b) all other integers L are such that $La_i \bmod(f)$ is very large (greater than f/h) for some i.

SOLVING SIII

We will use a lattice reduction algorithm due to Lenstra, Lenstra, and Lovacz [5] to solve SIII. The algorithm has the following properties

a) On Input

$$V_1, V_2, \ldots, V_n \text{ vectors in } R^n$$

Outputs

$V_1^*, V_2^*, \ldots, V_n^*$ vectors in R^n and integers $f_{i,j}$ $1 \leq i, j \leq n$ such that

1. $V_i^* = f_{i,1} V_1 + f_{i,2} V_2 + \ldots + f_{i,n} V_n \quad i = 1, 2, \ldots, n$

2. $|V_i^*| \leq 1.34^{\frac{n-1}{2}} |\lambda_i|$

   $i=1, 2, \ldots, n$

   where $|V|$ denotes Euclidian length of V and λ_i is the i^{th} successive minima of L (i.e., λ_i is the shortest vector in L which is not a linear combination of $\lambda_1, \lambda_2, \ldots, \lambda_{i-1}$).

b) It runs in polynomial time (independent of n).

Now consider the lattice L generated by the following vectors

$$\begin{aligned}
V_1 &= \langle \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_d, 0 \rangle \\
V_2 &= \langle f, 0, \ldots, 0, 0 \rangle \\
V_3 &= \langle 0, f, \ldots, 0, 0 \rangle \\
&\vdots \\
V_{d+1} &= \langle 0, 0, \ldots, f, 0 \rangle
\end{aligned}$$

where $\hat{a}_j$ is the j^{th} element in T.

As we have already argued the lattice L contains vectors

$$\begin{aligned}
W_1 &= \langle \hat{a}_1, \hat{a}_2, \ldots \hat{a}_d, 0 \rangle = 1.V_1 + 0.V_2 + \ldots + 0.V_{d+1} \\
W_2 &= \langle 2\hat{a}_1, 2\hat{a}_2, \ldots, 2\hat{a}_d, 0 \rangle = 2.V_1 + 0.V_2 + \ldots + 0.V_{d+1} \\
&\vdots \\
W_{\lfloor f/2^e a \rfloor} &= \langle \lfloor f/2^e a \rfloor \hat{a}, \lfloor f/2^e a \rfloor \hat{a}_2, \ldots, \lfloor f/2^e a \rfloor \hat{a}_d, 0 \rangle = \lfloor f/2^e \hat{a} \rfloor V_1 + 0.V_2 + \ldots + 0.V_{d+1} \\
U_0 &= \langle g_{1,1}, g_{1,2}, \ldots, g_{1,d}, 0 \rangle = \hat{L}_0 V_1 + k_1 V_2 + k_2 V_3 + \ldots + k_d V_{d+1} \\
&\vdots \\
U_z &= \langle g_{z,1}, g_{z,2}, \ldots, g_{z,d}\ 0 \rangle = \hat{L}_z V_1 + k_1 V_2 + k_2 V_3 + \ldots + k_d V_{d+1}
\end{aligned}$$

for some integers $\hat{L}_z$ (in fact $\hat{L}_z = \hat{L}_0 + z$) and where $|U_0| \leq \sqrt{d}\ f/2^e$.

Since all the W_i's are linear combinations of W_1 it follows that if L contains no other vector Y such that $|Y| \leq \sqrt{d}\ f/2^e$ then $\lambda_2 = U_0$.

Unfortunately, the lattice reduction algorithm is not guaranteed to find λ_2 (and therefore the desired k_i's) but is only guaranteed to find a vector V_2^* which is not to much longer.

$$|V_2^*| < 1.34^{\frac{d}{2}} |\lambda_2| \leq (1.34^{\frac{d}{2}})(\sqrt{d})(f/2^e).$$

If, however, we could guarantee that L contains no "exceptional" vector Y different than the W_i's and U_i's such that $|Y| \leq (1.34^{\frac{d}{2}})(\sqrt{d})(f/2^e)$ then the lattice reduction algorithm has no choice but to give us one of the U_i's as V_2^* and therefore we obtain the desired k_i's .

We have argued that by increasing d we reduce the probability that L contains such an "exceptional" vector. On the otherhand increasing d increases the "inaccuracy" of the lattice reduction algorithm. These opposing pressures will balance out favorably and we will with high probability obtain the k_i's when d is such that

$$\left(\frac{2^e}{(\sqrt{d})(1.34^{d/2})}\right)^d > 2m$$

or taking logs

$$(**)\quad ed - \frac{d}{2}\log(d) - \frac{d^2}{2}\log(1.34) > \log(m)+1$$

such a d will not exist if e is small with respect to m.

For example, if the b_i's are approximately 2^{200} and if m is approximately 2^{214} then for d = 31 the right hand side of (**) is about 216 and the left hand side is about 215. So that the attack described is very likely to succeed. However, if m is approximately 2^{213} then no appropriate d exists and the attack cannot be guaranteed to find the desired k_i's.

It is important to note that these calculations for d are based on the worst case running of the lattice reduction algorithm. Experience suggests that the average behavior of the algorithm, at least on cryptographically generated lattices, is far better. In fact, so much better that I believe it would be prudent, in the absence of countervailing information, for cryptographers to assume that the algorithm always finds exactly λ_2 (or at least misses by at most a polynomial in d). (See Lagarias [4] for conjectures in a different direction.

NOTICE

Heuristic arguments similar to those above are used to justify several of the steps which follow. Because these additional arguments involve no new ideas, the details will be omitted and only an outline will be provided. In general these arguments require showing that a given system has some "special" or expected solution and that under "reasonable randomness assumptions" other "exceptional" solutions can be made arbitrarily rare by increasing the number of inequalities in the system.

Returning to the attack we now assume that the k_i's have been found. Therefore, SII becomes

SVII

$$0 < La_i - k_iM \leq M/2^e \qquad i \in S$$

with properties:

P9 (Good) Among the solutions $< L, M >$ is the desired one $< \ell, m >$.

P10 (Good) The system is linear.

P11 (Bad) There are infinitely many undesirable solutions.

To overcome P11 we need a way to distinguish $< \ell, m >$ from the other solutions. We may do this by observing that what makes ℓ, m special is that at least for small i,

$$\ell a_i - k_i m = b_i = r_{h,i} * c_i * r_{\ell,i}$$

is not only less than $m/2^e$, but has a "window" of leading zeros in the high order bits of c_i. In other words, $\ell a_i - k_i m \bmod (2^{y+z})$ is small. Or equivalently, there are integers q_i such that $\ell a_i - k_i m - q_i 2^{y+z}$ is small.

Now consider the following system, where $\hat{T}$ is an h element subset of T, $\hat{S} = \{i | i \in \hat{T}\}$ and b is an integer which will be considered later.

SVIII

(A) $0 < La_i - k_iM \leq M/2^e \qquad i \in \hat{S}$

(B) $0 < La_i - k_iM - Q_i 2^{y+z} \leq 2^{y+z-b} \qquad i \in \hat{S}$.

This system has the following properties:

P12 (Good) It is linear.

P13 (Good) If $i \in \hat{S} \Longrightarrow c_i$ has a least b leading zeros then

a) Among the solutions $< L, M, < Q_i >_{i \in \hat{S}} >$ is the desired solution $< \ell, m, \{q_i\}_{i \in \hat{S}} >$

b) For h large enough, then with high probability all undesirable solutions are of one of the following forms:

i) $< L, M, \{Q_i\}_{i \in \hat{S}} >$ with $Q_i \neq 0$ and M much larger than m.

ii) $< L, M, \{0\}_{i \in \hat{S}} >$.

Consider for example $b = z/2$. Assume, as is reasonable, that $i < n/2 \Longrightarrow c_i$ has z/2 leading zeros. Then with probability $1/2^h$, $\hat{S}$ will contain only i's such that $i < n/2$ and a) will hold. Consider an $M \neq m$ such that $< L, M, \{Q_i\}_{i \in \hat{S}} >$ is a solution to SVIII (A).

Such M are sparse as can be seen from the "scaling" arguments of Shamir. For such M we would expect that $La_i - k_iM$ would have $1/2^{z/2}$ chance of having a window of z/2 zeros starting at bit y+z. Further the chances that for all $i \in \hat{S}$ $La_i - k_iM$ would have such a window would be expected to be $(1/2^{z/2})^h$. If h is large then M is extremely unlikely to satisfy SVIII(B) in addition to SVIII(A). The exception is when M is so small that $La_i - k_iM$ is actually less than $2^{y+z-z/2}$ and this gives a solution of type ii.

To give some idea of the selection of h assume M is about 2^{200}, z = 100, n = 100. Then if h = 4 we will choose an appropriate $\hat{S}$ after about 16 tries and a large $M \neq m$ which satisfies SVIII(A) will have about $1/2^{200}$ chance of also satisfying SVIII(B).

SOLVING SVIII

Again we will use the lattice reduction algorithm . Consider the lattice L generated by the following vectors, we assume b = z/2 and $\hat{T}$ contains only elements with a window of z/2 elements at the $y+z^{th}$ bit.

$$
\begin{array}{llllllllll}
V_1 & = < \hat{a}_1, & \hat{a}_2, & \ldots, & \hat{a}_h, & g\hat{a}, & g\hat{a}_2, & \ldots, & g\hat{a}_h & > \\
V_2 & = < \hat{k}_1, & \hat{k}_2, & \ldots, & \hat{k}_h, & g\hat{k}, & g\hat{k}_2, & \ldots, & g\hat{k}_h & > \\
V_3 & = < 0, & 0, & \ldots, & 0, & g2^{y+z}, & 0, & \ldots, & 0 & > \\
V_4 & = < 0, & 0, & \ldots, & 0, & 0, & g2^{y+z}, & \ldots, & 0 & > \\
 & & & \vdots & & & & & & \\
V_{h+2} & = < 0 & 0, & \ldots, & 0, & 0, & 0, & \ldots, & g2^{y+z} & >
\end{array}
$$

where $\hat{a}_j$ is the j^{th} element in $\hat{T}$

where $\hat{k}_j$ is the k corresponding to $\hat{a}_j$

and

where g is approximately $m/2^{y+z/2}$

(the purpose of g is to make sure that the constraints of SVIII(B) are not "lost" in solutions to SVIII(A) when the Euclidean metric is used. Notice for example, that when $W = \ell V_1 + mV_2 + q_1V_3 + \ldots + q_hV_{h+2}$ is considered then each entry in W is about m, whereas, without the g the last h entries of W would be negligible compared to the first h entries).

We know from the preceeding that the following vectors are in L

$$W = \ell . V_1 + mV_2 + q_1V_3 + q_2V_4 + \ldots + q_hV_{h+2}$$

where $|W|$ is approximately $\sqrt{2h}(m)$

$$U_j = \hat{L}_jV_1 + M_jV_2 + 0V_3 + 0V_4 + \ldots + 0V_{h+2}$$

for various j.

By the preceeding arguments we can be reasonably sure that all other vectors $Y \in L$ have $|Y|$ much larger than $|W|$.

Now since U_i are all dependent on V_1 and V_2 alone. Then will be dependent of λ_1, λ_2. The U's taken care of, that leaves W as λ_3 and no other vectors of nearly comparable smallness. Therefore the lattice reduction algorithm should find

$$V_3^* = W = \ell V_1 + mV_2 + q_1V_3 + q_2V_4 + \ldots + q_hV_{h+2}$$

and the desired ℓ and m have been recovered.

WARNING

The situation just considered is very complicated and cumbersome. The agruments presented are far from being proofs, and at best only provide partial justification for believing that the methods suggested are effective. Many things could go wrong. The ultimate test of these techniques is whether they work on real problems.

In the case of GSK's the method has been implemented by the author. On five different trials where roughly $z \simeq 10^{16}$ $y \simeq 10^{8}$, $m \simeq 10^{64}$, $w \simeq 10^{64}$ the method succeeded and w, m were recovered after only several hours of computation on a personal computer. To facilitate the trials the sets T and $\hat{T}$ and the numbers, e, d and, h, were pre-chosen so that the trial and error parts of the hueristic could be avoided.

In particular one trial used:

m	=	3710007477163539079884927556810340706993256827446844839469523587
w	=	2754665076473556947305356290417812811859660331309122185053365832
b_{i_1}	=	51330876000000000000327552192031
b_{i_2}	=	13235976000000000000918428068384
b_{i_3}	=	17399971000000000000524397343633
b_{i_4}	=	25002900572438811724397163537692

$$T = \{a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4}\}$$

$$\hat{T} = \{a_{i_1}, a_{i_2}, a_{i_3}\}$$

II. ITERATED KNAPSACK (IK)

Below we indicate how the techniques above might be used to attack the Iterated knapsack systems of Merkle and Hellman. Regretably, the author did not allocate adequate time for a detailed analysis and exposition of the ideas involved. Accordingly only a brief sketch is provided. The ultimate test of the efficacy of these ideas is of course whether they work on real problems.

Below we describe the procedure for generating a mated pair of keys for an IK:

STEP 0:

Generate positive integers z, n, y

Generate a sequence $a_{0,1}, a_{0,2}, \ldots, a_{0,n}$ with

$a_{0,i} \in \{0,1\}^z$, $i = 1, 2, \ldots, n$ such that

$$a_{0,i} \geq \sum_{j=1}^{i-1} a_{0,j} \qquad i = 2, 3, \ldots, n$$

STEP 1:

Generate positive integers w_1, m_1 such that

a) $(w_1, m_1) = 1$

b) $m_1 > \sum_{i=1}^{n} a_{0,i}$

Calculate

$$a_1, i \equiv w_1, a_{0,i} \mathrm{MOD}(m_1)$$

STEP Y:

Generate positive integers w_y, m_y such that

a) $(w_y, m_y) = 1$

b) $m_y > \sum_{i=1}^{n} a_{y-1,i}$

Calculate

$$a_i \equiv w_y a_{y-1,i} \mathrm{MOD}(m\ )$$

STEP Y+1

Generate a permutation π on $\{1, 2, \ldots, n\}$ output as the public key $< a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)} >$ keep $< w_1, m_1, w_2, m_2, \ldots, w_y, m_y, < a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)} >>$ as the private key.

We know that there are natural numbers $k_1, k_2, \ldots, k_n$ such that

$$\ell a_i - k_i\ m = b_i \qquad i = 1, 2, \ldots, n \quad .$$

We may assume that for an appropriate choice of e and of d element subset S of $\{1, 2, \ldots, n\}$ that, using the techniques of section II, we have recovered k_i, $i \in S$. We now consider the system

TI $\qquad 0 < La_i - k_i M \leq M/2^e \qquad i \in S$

TI has the following properties:

Q1 (Good) It is linear.

Q2 (Good) Among the solutions (L, M) is the "desired" one (ℓ, m).

Q3 (Bad) There are infinitely many undesirable solutions.

We therefore wish to find a means of distinguishing (ℓ, m) from other solutions. What makes (ℓ, m) special is that

$$(1) \quad \ell a_i - k_i\ m = b_i$$

and b_i is itself the result of a previous step in the key generation process. That is, there are $\hat{e}$, $\hat{\ell}$, $\hat{m}$ such that

$$\hat{\ell} b_i - \hat{k}_i\ \hat{m} = c_i$$

and

$$(2) \quad \hat{\ell} b_i - k_i\ \hat{m} \leq \hat{m}/2^{\hat{e}}$$

We will use tricks similar to those for obtaining the k_i's to obtain the $\hat{k}_i$'s. Consider the following

TII

$$\text{(A)} \quad 0 < La_i - k_iM \leq M/2^e \qquad i \in S$$

$$\text{(B)} \quad 0 < La_i - k_iM - \hat{K}_i f \leq f/2^e \qquad i \in S$$

when $f > 2^e b_i$, $i \in S$ and d is sufficiently large we should obtain from the corresponding lattice problem a $V_3{}^*$ whose representation as a linear combination of the inputs will give us $\{\hat{k}_i\}_{i \in S}$.

To see part of the reason for this consider multiplying (1) above by $\lfloor A \rfloor = \frac{f\hat{\ell}}{\hat{m}} + \varepsilon$ for positive ε less than 1 and multiplying (2) above by $\frac{f}{\hat{m}}$. From these equations it can be seen that $\{\lfloor A \rfloor \ell, \lfloor A \rfloor m, \{k_i\}_{i \in S}\}$ is a solution to TII.

having obtained the $\hat{k}_i$'s we finally consider the system

$$\text{(A)} \quad 0 < La_i - k_iM \leq M/2^e \qquad i \in S$$

$$\text{(B)} \quad 0 < La_i - k_iM - \hat{k}_i\hat{M} \leq \hat{M}/2^{\hat{e}} \qquad i \in S \ .$$

Solving this using the lattice reduction algorithm should give us a solution $< L, M, \hat{M} >$ with the property that

$$\frac{L}{(L,M)} = \ell \quad \text{and} \quad \frac{M}{(L,M)} = m \ .$$

Part of the reasoning here is that there are solutions to TIII of the form $< A\ell + g, Am + h, B >$ for integers A, B, g, h. But g,h should typically be very small with respect to A and B. In fact for very small A and B (e.g., $A = \hat{\ell}\ell$) g and h should be controllable and we should be able to find a solution to TIII where the gcd property holds. Failing this, at least a large portion of the high order bits of $\hat{\ell}, \ell, m, \hat{m}$ should be discovered.

ACKNOWLEDGEMENT

I would like to thank Jeff Lagarias, Ken Manders, Andrew Odlysko, Adi Shamir and Ron Rivest for their suggestions regarding this project.

REFERENCES

[1] W. Diffie, and N. Hellman, New Directions in Cryptography, IEEE Trans. Information Theory, IT-22-6, November, 1976.

[2] A. Lempel, Cyrptology in Transition: A Survey, Program 134-45-90, Discrete Mathematics Department, Digital Techniques Laboratory, Sperry Research Center (1978).

[3] Lagarias, J., Knapsack-Type Public Key Cryptosystems and Dcophantine Approximation, (abstract).

[4] J. Lagarias, The Computational Complexity of Simultaneous Dcophantine Approximation Problems, Proceedings 23rd Foundations of Computer Science Conference (1982) pg. 32.

[5] A.K. Lenstra, H.W. Lenstra, Jr., and L. Lovasz, Factoring Polynomials with Rational Coefficients, Report 82-05, Mathematics Institute, University of Amsterdam, March 1982.

[6] K.L. Manders and L. Adleman, NP-Complete Decision Problems for Binary Quadratics, J. Computer and Systems Science 16 (1978), 168-184.

[7] R. Merkle, N. Hellman, Hiding Information and Signatures in Trapdoor Knapsacks, IEEE Trans. Information Theory, IT-24-5, September, 1978.

[8] A. Shamir, A Polynomial Time Algorithm for Breaking Merkle-Hellman Cryptosystems, Proceedings 23rd Foundations of Computer Science Conference (1982).

[9] R. Rivest, A. Shamir and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, CACM 21-2, February, 1978.