SOLVING LOW DENSITY KNAPSACKS†

Ernest F. Brickell
Sandia National Laboratories
Albuquerque, New Mexico 87185

INTRODUCTION

Let $a_1,\ldots,a_n$ and s be a set of integers. The knapsack (or subset sum) problem is to find a 0-1 vector $(\varepsilon_1,\ldots,\varepsilon_n)$ such that $\Sigma\ \varepsilon_i a_i = s$ or to show that such a vector does not exist. The integers $a_1,\ldots,a_n$ are sometimes referred to as weights. The general knapsack problem is known to be NP complete [5,6]. Several cryptosystems based on the knapsack problem have been designed [9,12,16]. In April, 1982, Adi Shamir [14] announced a method for breaking the Merkle-Hellman cryptosystem. Since that time there has been a flurry of activity to extend his results to include all of the proposed knapsack based cryptosystems [1,2,3,7,13].

In a knapsack based cryptosystem, the cryptodesigner publishes a set of integers $a_1,\ldots,a_n$. A 0-1 vector $(\varepsilon_1,\ldots,\varepsilon_n)$ is encrypted by forming the sum $s = \sum_{i=1}^{n} \varepsilon_i a_i$. The cryptodesigner keeps secret certain information about the way the a_i's were chosen. This information allows him to decrypt any message, i.e., solve any knapsack problem where the integers $a_1,\ldots,a_n$ are the set of weights.

In all of the techniques mentioned above for breaking knapsack based cryptosystems, the cryptanalyst, using only the integers

† This work performed at Sandia National Laboratories supported by the U.S. Department of Energy under contract number DE-AC04-76DP00789.

$a_1,\ldots,a_n$, manages to find some of the secret information. In fact, the cryptanalyst can find enough information so that he also can solve any knapsack problem where the integers $a_1,\ldots,a_n$ are the set of weights.

Let $a_1,\ldots,a_n$ be a set of positive integers. Let $A = \max\{a_1,\ldots,a_n\}$. We define the *density* of a knapsack problem with weights $a_1,\ldots,a_n$ to be $n/\log_2 A$.

In this paper, we describe two methods for solving knapsacks of low density. Method 1 is a technique for solving knapsack problems that appears to work for almost all knapsacks of density less than $1/\log_2 n$. Method 2 is a slightly different technique that works on almost all knapsacks of density less than $d(n)$, where $d(n)$ is a function of n that is significantly larger than $1/\log_2 n$. However, our estimates of $d(n)$ come from emperical studies.

J. Lagarias and A. Odlyzko [8] have developed another technique for solving knapsacks of low density. The techniques are quite different. In our method, part of the algorithm works only with the weights $a_1,\ldots,a_n$ and runs in $O(n^4 (\log n)^3)$. If it is successful, then any knapsack problem in which the weights are $a_1,\ldots,a_n$ can be solved in $O(n^3)$. The Lagarias-Odlyzko method requires $O(n(\log A)^3)$ running time for each solution of a knapsack problem. They prove that their technique is expected to succeed if the density $\leq 1/n$ and show emperically that it is expected to succeed for much higher densities.

All of the above techniques for solving knapsack problems use the Lenstra-Lenstra-Lovasz (L^3) basis reduction algorithm for lattices [11]. A subset of points, L, of $\mathbf{R}^n$ is a lattice of rank n if $L = \{\sum_{i=1}^{n} z_i v_i : z_i \in \mathbf{Z}\}$ where $v_1,\ldots,v_n$ is some set of independent vectors in $\mathbf{R}^n$. The vectors $v_1,\ldots,v_n$ are said to be a basis of L. The L^3 algorithm finds a "short" or reduced basis for a given lattice. In [11], there are worst case bounds given on the lengths of vectors in a reduced basis. However, in emperi-