in Example 1. Let us choose $m = 61$, $a = 17$; then $b = 18$ and the enciphering key is $(34, 51, 58, 11, 39)$. To send the message 'WHY' our correspondent would compute 'W'$= (10110)_2 \mapsto 51 + 58 + 39 = 148$, 'H'$= (00111)_2 \mapsto 34 + 51 + 58 = 143$, 'Y'$= (11000)_2 \mapsto 11 + 39 = 50$. To read the message $148, 143, 50$, we first multiply by 18 modulo 61, obtaining $41, 12, 46$. Proceeding as in Example 2 with $V = 41$, $V = 12$, and $V = 46$, we recover the plaintext $(10110)_2$, $(00111)_2$, $(11000)_2$.

Of course, as usual there is no security using single-letter message units with such a small value of $k = 5$; Example 3 is meant only to illustrate the mechanics of the system.

For a while, many people were optimistic about the possibilities for knapsack cryptosystems. Since the problem of breaking the system is in a very difficult class of problems (NP-complete problems), they reasoned, the system should be secure.

However, there was a fallacy in that reasoning. The type of knapsack problem $C = \sum \epsilon_i w_i$ that must be solved, while not a superincreasing knapsack problem, is nevertheless of a very special type, namely, it is obtained from a superincreasing problem by a simple transformation, i.e., multiplying everything by a and reducing modulo m. In 1982, Shamir found an algorithm to solve this type of knapsack problem that is polynomial in k. Thus, the original Merkle–Hellman cryptosystem cannot be regarded as a secure public key cryptosystem.

One way around Shamir's algorithm is to make the knapsack system a little more complicated by using a sequence of transformations of the form $x \mapsto ax \ mod \ m$ for different a and m. For example, we might simply use two transformations corresponding to (a_1, m_1) and (a_2, m_2). That is, we first replace our superincreasing sequence $\{v_i\}$ by $\{w_i\}$, where w_i is the least positive residue of $a_1 v_i \ mod \ m_1$, and then obtain a third sequence $\{u_i\}$ by taking the least positive residue $u_i = a_2 w_i \ mod \ m_2$. Here we choose random m_1, m_2, a_1 and a_2 subject to the conditions $m_1 > \sum v_i$, $\quad m_2 > k m_1$, and $g.c.d.(a_1, m_1) = g.c.d.(a_2, m_2) = 1$. The public key is then the k-tuple of u_i, and the enciphering function is $C = f(P) = \sum_{i=0}^{k-1} \epsilon_i u_i$, where $P = (\epsilon_{k-1} \cdots \epsilon_1)_2$. To decipher the ciphertext using the key $K_D = (b_1, m_1, b_2, m_2)$ (where $b_1 = a_1^{-1} \ mod \ m_1$ and $b_2 = a_2^{-1} \ mod \ m_2$), we first compute the least positive residue of $b_2 C$ modulo m_2, and then take the result, multiply it by b_1, and reduce modulo m_1. Since $b_2 C \equiv \sum \epsilon_i w_i \ mod \ m_2$, and since $m_2 > k m_1 > \sum w_i$, it follows that the result of reducing $b_2 C \ mod \ m_2$ is *equal* to $\sum \epsilon_i w_i$. Then when we take $b_1 \sum \epsilon_i w_i \ mod \ m_1$ we obtain $\sum \epsilon_i v_i$, from which we can determine the ϵ_i using the above algorithm for a superincreasing knapsack problem.

At the present time, although there is no polynomial time algorithm which has been proved to give a solution of the iterated knapsack problem (i.e., the public key cryptosystem described in the last paragraph), Shamir's algorithm has been generalized by Brickell and others, who show that iterated knapsack cryptosystems are vulnerable to efficient cryptanalysis. In