# Knapsack System

## 1 Subset Sum

The **Subset sum** problem goes as follows. Let

$$s_1, s_2, \ldots, s_n$$

be $n$ positive integers called *sizes*, and let $T$ be another positive integer called *target*. The problem consists in finding (if possible) a $0-1$ vector

$$(x_1, x_2, \ldots, x_n)$$

such that

$$T = \sum_{i=1}^{n} x_i s_i.$$

In general this problem has been shown to be "**NP-complete**". This is an important notion of theoretical computer science that implies (among a lot of other things) that there is no known polynomial-time algorithm that solves it.

However, some special cases of this problem are very easy to solve. This is the case for "superincreasing" sequences. A list of sizes is said to be superincreasing if for each $j$ from 2 to $n$, one has

$$s_j > \sum_{i=1}^{j-1} s_i.$$

To check if $T$ can be expressed as a subset sum of the $s_i$'s, one can proceed as follows.

> **If** $T > \sum_{i=1}^{n} s_i$ **or** $T < s_1$ **then** there is no solution.
> **Else** find the largest $k$ such that $s_k \leq T$, **set**
>
> $$x_k := 1, \qquad \text{and} \qquad x_j := 0 \quad (\text{for } j > k).$$
>
> **and** recursively find how $T - s_k$ can be expressed as a subset sum of the $s_i$'s.

**Example.** Suppose that the chosen superincreasing sequence is

$$2, 5, 9, 21, 45, 103, 215, 450, 946$$

and $T = 1643$. Then

$$\begin{aligned}
T &= (T-946)+946 \\
&= (697)+946 \\
&= (697-450)+450+946 \\
&= (247)+450+946 \\
&= (247-215)+215+450+946 \\
&= (32)+215+450+946 \\
&= (32-21)+21+215+450+946 \\
&= (11)+21+215+450+946 \\
&= (11-9)+9+21+215+450+946 \\
&= (2)+9+21+215+450+946 \\
&= 2+9+21+215+450+946
\end{aligned}$$

It is easy to build a "random" superincreasing sequence as follows. For a fixed $k > 1$, and $n$ being the intended length of the sequence,

> **If** $n = 1$ **then** choose $s_1$ at random between 1 and $k$.
> **Else** recursively build a superincreasing sequence of length $n-1$, say
>
> $$(s_1, s_2, \ldots, s_{n-1})$$
>
> **and** extend it by adding a last size $s_n = m + j$, where
>
> $$m = \sum_{i=1}^{n-1} s_i,$$
>
> and $j$ is chosen at random between 1 and $k$.

## 2 Merkle-Hellman Knapsack Cryptosystem

To make a public key system out of the Knapsack (Subset sum) problem, one proceeds as follows.

1. Choose
$$\mathbf{s} = (s_1, s_2, \ldots, s_n)$$
a superincreasing list of integers with $n$ large, and choose $p$ be a large prime such that
$$p > \sum_{i=1}^{n} s_i.$$

2. Let $a$ be a random number between 1 and $p-1$, and set
$$t_i := a\, s_i \mod p.$$

The vector $\mathbf{t} = (t_1, t_2, \ldots, t_i)$ is then made public. To encode a message $(x_1, x_2, \ldots, x_n)$ (made of bits of 0 and 1), one sends the single number

$$C := \sum_{i=1}^{n} x_t\, t_i.$$

To decode, we need only solve the subset sum problem for $M := (a^{-1}\, C \bmod p)$, with the sequence $\mathbf{s}$ known to be superincreasing, since it is clear that

$$M = \sum_{i=1}^{n} x_t\, s_i.$$

# 3 Breaking

This system has been broken since the 1980's, but a variation is still not broken. The key to making it "unbreakable" resides in the choice of the transformation from $\mathbf{s}$ to $\mathbf{t}$. This transformation must make the resulting subset sum problem, for the new *sizes* (entries of $\mathbf{t}$), hard to solve.

**Exercises.**

1. Decode 6665 knowing that $p = 2003$, $a = 1289$, and using the superincreasing sequence

$$\mathbf{s} = (2, 5, 9, 21, 45, 103, 215, 450, 946).$$

2. Suppose we use a Knapsack encryption system with sequence

$$s = \{1, 4, 11, 23, 48\}$$

with modulus $m = 101$ and multiplier $a = 9$.

   (a) Encrypt the message 1 0 1 0 1
   (b) Decrypt the message 76.