Today
- Merkle–Hellman Knapsack Cryptosystem
  - Shamir's Secret Sharing Scheme

Knapsack Cryptosystems

Idea: message encoding turns easy knapsack problem into hard one

Variant of knapsack (actually subset-sum) problem:
Given $n$ numbers $s_1, \ldots, s_n$ and target $t$
find subset which sums to $t$ (if one exists)

Problem is NP-hard in general, but not when $s_1 \cdot s_n$ is superincreasing

Definition: A sequence $s_1, \ldots, s_n$ is superincreasing if $s_i > \sum_{j<i} s_j \ \forall i$

Given target $t$ and superincreasing sequence, knapsack easy to solve

Why? (Greedy alg, starting with largest $s_i$)

Merkle–Hellman Cryptosystem ['78]

Private: superincreasing $s_1, s_2, \ldots, s_n$
  integer $M$
  integer $a$ (1 $a$ $M$) s.t. gcd($M, a$) = 1

Public: sequence $t_1, t_2, \ldots, t_n$ where $t_i = a s_i \mod M$

Encryption:

Plaintext is n-bit string $b_1, b_2, \ldots, b_n$

Cipher-text = $\sum_{i=1}^n b_i t_i$

Decryption:

Multiply ciphertext by $a^{-1} \mod M$ to get $T$

Find subset of $s_1, s_2, \ldots, s_n$ summing to $T$
Example:

Private Key  S = \{2, 3, 6, 13, 27, 52\}  M = 105 , a = 31

Public Key  S' = \{62, 93, 81, 88, 102, 37\}

Message  101110

Ciphertext  = 62 + 81 + 88 + 102 = 333

Recipient knows  a^x = 61 (since 31 \cdot 61 = 1 \mod 105)

Decodes  333 \cdot 61 = 48 \mod 105 \rightarrow 48 = 27 + 13 + 6 + 2 , i.e. 10110

Cool scheme, but broken in early 80's!

Knapsack is NP-hard in worst case, not all instances are hard.

Technique to break system called Lattice-Basis-Reduction.

Idea: Consider (n+1)x(n+1) matrix

\[
\begin{pmatrix}
100 & 0 & \cdots & 0 & -1 \\
010 & 0 & \cdots & 0 & -1 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
000 & 0 & \cdots & 1 & -1 \\
000 & 0 & \cdots & 0 & T
\end{pmatrix}
\]

LLL (Lenstra-Lenstra-Lovasz) algorithm will find a small-weight integer linear combo of the rows.

Note that \langle b_1, b_2, \ldots, b_n, 1 \rangle is such a small-weight vector, since all \(b_i\)'s are either 0 or 1. In fact, it's often the only non-trivial small-weight vector (this can be made precise for random knapsack problems).

Thus LLL-algorithm is likely to find it...

(finding shortest vector NP-hard in general, easier when one vec. very small)
Secret Sharing

How can we distribute a secret among n people so that any k or more of them can discover it? (motivation: nuclear key codes)

Simple example: If k=n, just give each person a random integer mod p (where p is a large int) and let the secret be the sum of all those integers mod p.

Shamir’s Threshold Scheme

- Fix a large prime p > n (p can be public knowledge)
- Choose a random polynomial of degree k-1 over \( \mathbb{Z}_p \)
  \[ p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} \quad a_0, a_1, \ldots, a_{k-1} \in \mathbb{Z}_p \]
- Choose n distinct non-zero points \( x_1, x_2, \ldots, x_n \)

Secret is value \( a_0 \)

Person i receives \( p(x_i) \) for \( i = 1, \ldots, n \)

Correctness

- Any k people can recover \( a_0 \) (just do polynomial interpolation to get all of \( a_0, a_1, \ldots, a_{k-1} \), or use Lagrange’s formula)

- Since \( a_0 = p(0) \), any group of \( \leq k \) people cannot recover \( a_0 \), since any value of \( p(0) \) will be consistent with some polynomial matching the values they have)