Merkle-Hellman Cryptosystem

Merkle-Hellman Cryptosystem

- The first algorithm for generalized public-key encryption was the knapsack algorithm developed by Ralph Merkle and Martin Hellman.
- Knapsack algorithms get their security from the knapsack problem, an NP-complete problem.
- The knapsack problem is a simple one. Given a pile of items, each with different weights, is it possible to put some of those items into a knapsack so that the knapsack weighs a given amount?
- More formally:
  - Given a set of values $M_1, M_2, \ldots, M_n$, and a sum S
  - Compute the values of bi such that $S=b_1M_1+b_2M_2+\ldots+b_nM_n$.
  - The values of b_i can be either zero or one.
  - A one indicates that the item is in the knapsack; a zero indicates that it isn't.
- For example:
  - The items might have weights of 1, 5, 6, 11, 14, and 20.
  - You could pack a knapsack that weighs 22; use weights 5,6, and 11.
  - You could not pack a knapsack that weighs 24.
- In general, the time required to solve this problem seems to grow exponentially with the number of items in the pile.

Merkle-Hellman Cryptosystem

- The idea behind the Merkle-Hellman knapsack algorithm is to encode a message as a solution to a series of knapsack problems. A block of plaintext equal in length to the number of items in the pile would select the items in the knapsack (plaintext bits corresponding to the b values), and the ciphertext would be the resulting sum.

Plaintext:	111 0 0 1	0 10 1 1 0	000 0 0 0	0 11 0 0 0
Knapsack:	156 11 14 20	156 11 14 20	156 11 14 20	156 11 14 20
Ciphertext:	1+5+6+20=	5+11+14=	0=	5+6=
	32	30	0	11

- The trick is that there are actually two different knapsack problems, one solvable in linear time and the other believed not to be.
  - The easy knapsack can be modified to create the hard knapsack.
  - The public key is the hard knapsack, which can easily be used to encrypt but cannot be used to decrypt messages.
  - The private key is the easy knapsack, which gives an easy way to decrypt messages.
  - People who don’t know the private key are forced to try to solve the hard knapsack problem.

Superincreasing Knapsacks

A superincreasing sequence is a sequence in which every term is greater than the sum of all the previous terms.

- For example, (1,3,6,13,27,52) is a superincreasing sequence,
- but (1,3,4,9,15,25) is not.

Superincreasing knapsack problem is easy to solve.

1. Take the total weight and compare it with the largest number in the sequence.
2. If the total weight is less than the number, then it is not in the knapsack.
3. If the total weight is greater than or equal to the number, then it is in the knapsack.Reduce the weight of the knapsack by the value.
4. Repeat until finished.
5. If the total weight has been brought to zero, then there is a solution. If the total weight has not, there isn’t.

Superincreasing Knapsacks

- Example: consider a total knapsack weight of 70 and a sequence of weights of (2,3,6,13,27,52).
  - The largest weight, 52, is less than 70, so 52 is in the knapsack. Subtracting 52 from 70 leaves 18.
  - The next weight, 27, is greater than 18, so 27 is not in the knapsack.
  - The next weight, 13, is less than 18, so 13 is in the knapsack. Subtracting 13 from 18 leaves 5.
  - The next weight, 6, is greater than 5, so 6 is not in the knapsack.
  - Continuing this process will show that both 2 and 3 are in the knapsack and the total weight is brought to 0, which indicates that a solution has been found.
- Were this a Merkle-Hellman knapsack encryption block, the plaintext that resulted from a ciphertext value of 70 would be 110101.

Merkle-Hellman Cryptosystem

- Superincreasing knapsack problem is easy to solve.
- Non-superincreasing, or normal, knapsacks are hard problems .The fastest algorithms, taking into account the various heuristics, grow exponentially with the number of possible weights in the knapsack.
- The Merkle-Hellman algorithm is based on this property.
- The private key is a sequence of weights for a superincreasing knapsack problem.
- The public key is a sequence of weights for a normal knapsack problem with the same solution.
- Merkle and Hellman developed a technique for converting a superincreasing knapsack problem into a normal knapsack problem.
- They did this using modular arithmetic.

Creating the Public Key from the Private Key

To get a normal knapsack sequence

1. Take a superincreasing knapsack sequence, for example (2,3,6,13,27,52)
2. Multiply all of the values by a number n, mod m.
3. The modulus should be a number greater than the sum of all the numbers in the sequence: for example, 105.
4. The multiplier should have no factors in common with the modulus: for example, 31.

- The normal knapsack sequence would then be

2 *31 mod 105 = 62	3*31mod105=93
6*31mod105=81	13*31mod105=88
27*31 mod 105 = 102	52*31 mod 105 = 37

- The knapsack would then be (62,93,81,88,102,37).
- Then

The public key is (62,93,81,88,102,37)

and the private key is (2,3,6,13,27,52)

Encryption

- To encrypt a binary plaintext,
  1. Break it up into blocks equal to the number of items in the knapsack sequence.
  2. Allowing a one to indicate the item is present and a zero to indicate that the item is absent,
  3. Compute the total weights of the knapsacks, one for every plaintext block.
- Example:

  plaintext = 011000 110101 101110

  public key = (62,93,81,88,102,37)

  private key = (2,3,6,13,27,52)

  – 011000 corresponds to 93 + 81 = 174
  – 110101 corresponds to 62 + 93 + 88 + 37 = 280
  – 101110correspondsto62+81+88+102=333

  The ciphertext would be 174,280,333

Decryption

- To decrypt the ciphertext,
  1. Determine n^{-1} such that $n(n^{-1}) = 1 \pmod{m}$.
  2. Multiply each of the ciphertext values by $n^{-1} \bmod m$
  3. Partition with the private knapsack to get the plaintext values.

- In our example:
  – The superincreasing knapsack is (2,3,6,13,27,52]
  – m is equal to105,
  – n is equal to 31, then n^{-1} is equal to 61
- The ciphertext message is 174,280,333.
  – 174 * 61 mod 105 = 9 = 3 + 6, which corresponds to 011000
  – 280 * 61 mod 105 = 70 = 2 + 3 + 13 + 52, which corresponds to 110101
  – 333 * 61 mod 105 = 48 = 2 + 6 + 13 + 27, which corresponds to 101110
- The recovered plaintext is 011000 110101 101110.

Security of Merkle-Hellman Cryptosystem

- Shamir and Zippel found flaws in the transformation that allowed them to reconstruct the superincreasing knapsack from the normal knapsack.
- The exact arguments are beyond the scope of this course.