

## MCS 320 Project One : Lattice Basis Reduction and the Knapsack Cryptosystem due Monday 26 September 2005, at 2PM.

The primary goal of this project is to use the modeling capabilities of Maple, applying its extended arithmetic to study a public key cryptosystem based on the hardness of the subset sum problem, a variant of the common knapsack problem. As our secondary goal, we will encounter an important algorithm in symbolic computation: the LLL-algorithm (the LLL stands for A.K. Lenstra, H.W. Lenstra, and L. Lovász) for lattice basis reduction. The LLL-algorithm — which runs in polynomial time — compromises the security of the knapsack cryptosystem.

### 0. Some useful Maple commands

This project description is accompanied by a Maple worksheet, available for downloading from the web site of this course. This worksheet describes at a more technical level the operations you need to simulate the knapsack cryptosystem. The sections in the worksheet are

**Encoding and Decoding Strings:** Our secret message is a string of characters. To compute with the message, we use the `convert` command to encode the message into a sequence of bytes. As every byte is a sequence of bits, our message is encoded into a list of lists of bits. To decode a list of lists of bits, the bit sequences are evaluated into bytes and then converted back into strings.

**The Subset Sum Problem for Super Increasing Sequences:** The subset sum problem is easy for a special sequence of weights (made precise in the section below). The worksheet provides functions to pack a sequence of bytes into a sequence of sums. In the opposite direction, we solve the subset sum problem to unpack a sequence of sums into a sequence of bytes.

**Encrypting and Decrypting Lists of Lists of Bits:** The encryption function uses a public key to encrypt a message encoded as a sequence of bytes. The public key is a sequence of weights, known to every one who wishes to send a message to a particular participant in the cryptosystem. With the secret key, known only to the receiver of the message, the public key is transformed into a super increasing sequence of weights, and the subset sum problem is then easy to solve, so the message is then decrypted into a string of characters. The worksheet contains actual numbers illustrating the working of a particular knapsack cryptosystem.

**Lattice Basis Reduction:** The LLL algorithm is the procedure `LLL` in the package `IntegerRelations`. In Maple 7, the command `lattice` works just as well. An example in the worksheet shows how to convert the subset sum problem into a lattice basis reduction.

To start the project, you must download the worksheet and execute it. Once you understand how the worksheet works, you are already halfway in your solution of this project.

### 1. The Knapsack Cryptosystem

The knapsack problem is familiar to every traveler: find a selection of objects with given weights whose sum of weights does not exceed the capacity of the knapsack. We consider a slightly more restrictive variant of this problem, the subset sum problem: given  $n$  weights  $\mathbf{w} = (w_1, w_2, \dots, w_n)$  and a sum  $s$ , find a bit vector  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$  so that  $\mathbf{w} \cdot \mathbf{x} = s$ , where  $\mathbf{w} \cdot \mathbf{x} = w_1x_1 + w_2x_2 + \dots + w_nx_n = s$ . Even as the weights are natural numbers, this is a computationally hard problem. The security of the knapsack cryptosystem is based on the hardness of this subset sum problem.

The subset sum problem is easy to solve for super increasing sequences of weights. The sequence  $\mathbf{w} = (w_1, w_2, \dots, w_n)$  is super increasing if  $w_k > w_1 + w_2 + \dots + w_{k-1}$ , for all  $k = 2, 3, \dots, n$ . For example, take  $w_k = 2^k$ , then the solution  $\mathbf{x}$  is just the binary expansion of the sum  $s$ .

To conceal a super increasing sequence  $\mathbf{w}$  we proceed as follows. Generate two numbers  $u$  and  $v$  such that  $\gcd(u, v) = 1$  and  $u > w_1 + w_2 + \dots + w_n$ . Compute  $\mathbf{a} = (a_1, a_2, \dots, a_n)$  as  $a_i = v \cdot w_i \bmod u$ , for  $i = 1, 2, \dots, n$ , or shortly  $\mathbf{a} = v\mathbf{w} \bmod u$ . The sequence  $\mathbf{a}$  is the public key, known to every one. The secret key consists of  $(u, v, \mathbf{w})$ .

The encryption of a secret  $\mathbf{x}$  is  $s = \mathbf{a} \cdot \mathbf{x}$ . Since  $\mathbf{a}$  is not super increasing, the subset sum problem is hard to solve and the decryption can be done easily only by using the secret key  $(u, v, \mathbf{w})$ . Let  $v^{-1} = (1/v) \bmod u$ . Then  $v^{-1}s \bmod u = v^{-1}\mathbf{a} \cdot \mathbf{x} \bmod u = v^{-1}v\mathbf{w} \cdot \mathbf{x} \bmod u = \mathbf{w} \cdot \mathbf{x} \bmod u$ . This is now a subset sum problem with a super increasing sequence  $\mathbf{w}$  and thus easy to solve.

## 2. Lattice Basis Reduction

A lattice is similar to a vector space. Like a vector space, a lattice is spanned (or generated) by a finite number of basis vectors. Unlike a vector space, the basis vectors have only integer entries and only integer linear combinations of the basis vectors are allowed.

Given any basis of a lattice, the LLL-algorithm finds a reduced basis, consisting of short vectors. A short vector has small length and its entries are small numbers, like the entries in the vectors of the subset sum problem are either zero or one.

Consider the following weights:  $\mathbf{w} = (366, 385, 392, 401, 422, 437)$  and sum  $s = 1215$ . This leads to the lattice basis  $(1, 0, 0, 0, 0, 0, -366)$ ,  $(0, 1, 0, 0, 0, 0, -385)$ ,  $(0, 0, 1, 0, 0, 0, -392)$ ,  $(0, 0, 0, 1, 0, 0, -401)$ ,  $(0, 0, 0, 0, 1, 0, -422)$ ,  $(0, 0, 0, 0, 0, 1, -437)$ , and  $(0, 0, 0, 0, 0, 0, 1215)$ . The output of the LLL-algorithm gives a reduced basis, which contains the vector  $(0, 0, 1, 1, 1, 0, 0)$  which solves the subset sum problem  $s = \mathbf{w} \cdot \mathbf{x}$ .

## 3. Assignments

To solve this project, you may use either the Maple 9.5 in the labs, or Maple 10. The four assignments below are relatively independent from each other, you can make progress simultaneously in all four assignments. Assignments one and two are linked and can be solved in any order. Assignments three and four are independent from each other, but should be solved after the first two assignments.

### 3.1 Assignment One: make your own worksheet model

One of the major strengths of Maple is the interaction between formulas and calculations. With proper documentation in a worksheet, we follow the same style rules as writing a technical report. When executed, the report functions as a computer program. The goal of this assignment is that you integrate the mathematical content of this note with the calculations you have to perform to solve the other three assignments. The end result should look as a self contained mathematical report. Currently, the mathematics of the note are disjoint from the Maple commands.

When implementing the mathematical model into a Maple worksheet, you could simultaneously solve the second assignment. The goal of assignment one is to incorporate the mathematical concepts into a Maple worksheet. For this assignment, you can still use the numbers in the example worksheet you download from the web site. Assignment two asks you to pick different numbers.

### 3.2 Assignment Two: build your own cryptosystem

The worksheet contains a very small example of numbers to simulate the knapsack cryptosystem. The goal of this assignment is that you pick your own numbers – make sure to pick them random enough so they are different from your neighbor! – and walk through all the steps in the encryption and decryption algorithms of your own particular message. Your message should be one paragraph of text of your choosing.

This assignment can be solved before solving assignment one by changing the numbers in the given worksheet. After solving assignment two, you should then rearrange the operations and integrate the mathematical model explained in this note into the worksheet.

### 3.3 Assignment Three: small numbers are insecure

By the LLL-algorithm, we can decrypt rather easily parts of the encrypted message. Show that by choosing larger numbers in public and secret keys the short vectors no longer lead to useful information. Illustrate this by giving the percentage of characters in your chosen paragraph (from assignment two) which can be successfully decrypted using LLL, with small numbers (for example the ones in the example worksheet) and with larger numbers (the ones you picked in assignment two).

The goal of this assignment is to show that with sufficiently large numbers, the security of the knapsack system increases.

### 3.4 Assignment Four: little public keys are insecure

The example worksheet has public keys of length 8, corresponding to the number of bits in one byte. This is very insecure, as one could try by brutal force all possible  $2^8$  combinations and thus crack the cryptosystem with relatively little effort. Doubling the length of the keys to 16 makes this brutal force attack harder.

Indicate which operations of the worksheet must be changed to work with longer public keys. For example, illustrate this on a message of three bytes, using a public key of size 24.

Estimate the cost of a brutal force attack, expressing the cost in function of the length of the public key. Give some examples, for various sizes of the keys, e.g.: if there are 100 numbers in the keys, it takes so many operations for one check, performed in so many milliseconds, multiplying this with the number of checks, decrypting the message would take so many years.

## 4. The deadline is Monday 26 September 2005 at 2PM

Bring *your* solution to the project to class. The *your* is emphasized to stress that your solution is the result of an *individual* effort. Collaborations are **not** permitted.

The solution to this project consists in two parts:

1. A print out of the Maple worksheet that you bring to class.

*Deliver a well written document, with grammatically correct and complete sentences, without spelling mistakes, appropriately structured into sections and subsections.*

2. The Maple worksheet that you eMail as an attachment to me.

*The worksheet should run as a computer program, from top to bottom with consistent output and without errors.*

If you have questions or difficulties with the assignments, feel free to come to my office for help.