

Instructions

1. You are encouraged to work in groups of two. If you cannot find a partner, you can work alone.
2. Please submit **one** copy of the assignment; if you are working with a partner, both names should appear on the assignment.
3. For **Part A** of the assignment, you must submit an electronic copy of your Perl application using subversion. Note that you will get a grade of zero if your program does not compile.

Part A

The *Merkle-Hellman subset-sum cryptosystem* works as follows. First Alice creates a secret key consisting of the following elements:

- A *super-increasing* sequence: $\mathbf{r} = (r_1, r_2, \dots, r_n)$ where $r_i \in \mathbb{N}$, and the property of being “super-increasing” refers to $2r_i \leq r_{i+1}$, for all $1 \leq i < n$.
- A pair of positive integers A, B with two conditions: $2r_n < B$ and $\gcd(A, B) = 1$.

The public key consists of $\mathbf{M} = (M_1, M_2, \dots, M_n)$ where $M_i = Ar_i \pmod{B}$.

Suppose that Bob wants to send a plain-text message $x \in \{0, 1\}^n$, i.e., x is a binary string of length n . Then he uses Alice’s public key to compute $S = \sum_{i=1}^n x_i M_i$, where x_i is the i -th bit of x , interpreted as integer 0 or 1. Bob now sends S to Alice.

For Alice to read the message she computes $S' = A^{-1}S \pmod{B}$, and she solves the subset-sum problem S' using the super-increasing \mathbf{r} . The subset-sum problem, for a general sequence \mathbf{r} , is very difficult, but when \mathbf{r} is super-increasing (note that \mathbf{M} is assumed not to be super-increasing!) the problem can be solved with a simple greedy algorithm.

More precisely, Alice finds a subset of \mathbf{r} whose sum is precisely S' . Any subset of \mathbf{r} can be identified with a binary string of length n , by assuming that x_i is 1 iff r_i is in this subset. Hence Alice “extracts” x out of S' .

For example, let $\mathbf{r} = (3, 11, 24, 50, 115)$, and $A = 113$, $B = 250$. Check that all conditions are met, and verify that $\mathbf{M} = (89, 243, 212, 150, 245)$. To send the secret message $x = 10101$, we compute $S = 1 \cdot 89 + 0 \cdot 243 + 1 \cdot 212 + 0 \cdot 150 + 1 \cdot 245 = 546$. Upon receiving S , we multiply it times 177, the inverse of 113 in mod 250, and obtain 142. Now x may be extracted out of 142 with a simple greedy algorithm.

Your task is to implement the Merkle-Hellman subset-sum cryptosystem in perl. Call the program `sscrypt`, and it should work with three different switches: `-e -d -v`, for *encrypt*, *decrypt* and *verify*. That is,

```
sscrypt -e M1 M2 ... Mn x
```

encrypts the string $x = x_1x_2 \dots x_n \in \{0, 1\}^n$ with the public key $\mathbf{M} = (M_1, M_2, \dots, M_n)$, and outputs S . On the other hand,

```
sscrypt -d r1 r2 ... rn A B S
```

decrypts the string $x = x_1x_2 \dots x_n \in \{0, 1\}^n$ from S using the secret key $\mathbf{r} = (r_1, r_2, \dots, r_n)$ and A, B ; that is, it outputs x on input \mathbf{r}, A, B, S . Finally,

```
sscrypt -v r1 r2 ... rn A B
```

checks that $\mathbf{r} = (r_1, r_2, \dots, r_n)$ is super-increasing, that $2r_n < B$ and that $\gcd(A, B) = 1$, and outputs the corresponding public key $\mathbf{M} = (M_1, M_2, \dots, M_n)$.

Part B

1. Show that if $\mathbf{r} = (r_1, r_2, \dots, r_n)$ is a super-increasing sequence then $r_{i+1} > \sum_{j=1}^i r_j$, for all $1 \leq i < n$.
2. Suppose that $\mathbf{r} = (r_1, r_2, \dots, r_n)$ is a super-increasing sequence, and suppose that there is a subset of \mathbf{r} whose sum is S . Provide a (natural) greedy algorithm for computing this subset, and show that your algorithm is correct.