

## Assignment 2

Due: 14/10/2008

1. (i) It is well known that if a positive integer  $n$  is expressed in the usual base 10 notation, then  $n$  is divisible by 9 if and only if the sum of its decimal digits is a multiple of 9. Prove this result. [Hint: Consider  $n - s$ , where  $s$  is the sum of the digits.]  
(ii) Let  $n = (123456713572461473625152637416427531765432)_8$  (expressed in octal notation.) Is  $n$  a multiple of 7?
2. Find the smallest integer  $n$  such that  $\phi(n) < 0.192n$  (where  $\phi$  is Euler's phi function). To obtain full marks you must show that your answer is correct.
- \*3. (MATH2988) A positive integer  $n$  is called a *Sierpinski number* if  $2^k n + 1$  is composite for all natural numbers  $k$ . The smallest  $n$  definitely known to be a Sierpinski number is 78557. Prove that for each natural number  $k$  there exists  $p \in \{3, 5, 7, 13, 19, 37, 73\}$  such that  $p$  is a factor of  $2^k \times 78557 + 1$ .

The remaining questions are to be done using magma. After logging in, start a magma session, and type

```
SetLogFile("asst2.txt");  
load "a2data.txt";
```

**You must type the SetLogFile command before loading the data file.**

After a2data.txt has been loaded, magma will have to do some preliminary calculations which may take a little while.

When it is ready for you, answer the questions, and after ending your magma session hand in the file asst2.txt by clicking the "handin" icon on your desktop, selecting m2068asst2, and then selecting the file asst2.txt.

Alternatively, you may use the Computer Tutorial Log File Upload page to submit your file.

You may (if you wish) edit the file asst2.txt before handing it in; e.g. you could use Crimson Editor to remove incorrect lines. But you must not delete the lines that were printed initially when asst2.m was loaded, and you must ensure that the file you submit contains all the correct magma commands needed to answer the questions, and Magma's responses to these commands.

4. Loading `a2data.txt` caused magma to compute a number  $m$  that is the product of two distinct prime numbers  $a$  and  $b$ , and also to compute  $\phi(m) = (a-1)(b-1)$ . Type `m`; and `phi`; to see the values of  $ab$  and  $\phi(ab)$ . The numbers  $a$  and  $b$  were deleted after `m` and `phi` were computed. Your task is to recover them by solving an appropriate equation. [Hint: See Question 1 of Tutorial 7 and Question 8 of Computer Tutorial 9.]

5. The number  $p = 1084744631483$  is prime, and 13 is a primitive root mod  $p$ . Enter the command

```
FF:=FiniteField(1084744631483);
```

thereby defining `FF` to be the set of residues modulo  $p$  and telling magma to use residue arithmetic for computations with elements of `FF`. Type `b:=FF!13;`, defining `b` to be 13 regarded as an element of `FF`.

Type `a:=Random(FF);`, telling magma to randomly choose a residue mod  $p$  and call it `a`. Type `a`; to see the number it has chosen. Then find a number `k` such that  $b^k$  equals `a` in `FF`. [This can be done with a single magma command.]

6. When `a2data.txt` was loaded magma set up an Elgamal cryptosystem for you to use. Your private key is `mm`, and your public key consists of a prime `pp`, a number `bb` that is a primitive root modulo `pp`, and `kk` which is the residue of  $bb^{mm}$  modulo `pp`. Before proceeding further, get magma to print out the values of these four numbers (via the commands `mm;`, `pp;`, `bb;` and `kk;`).

You receive a message that has been enciphered using your public key. The message consists of two parts, `ct[1]` and `ct[2]`. The first part is the number `Modexp(bb,i,pp)` where `i` is a number randomly chosen by the person who sent you the message. The second part, `ct[2]`, is a sequence of residues modulo `pp`. It was created by taking the original plaintext, encoding it as a sequence of residues mod `pp`, and multiplying each term by the scrambling factor `Modexp(kk,i,pp)` (using residue arithmetic mod `pp`) to produce the sequence `ct[2]`.

(i) Determine the scrambling factor, and its inverse modulo `pp`.

(ii) Determine the encoded plaintext.

(iii) Use the `NaiveDecoding` function to obtain the unencoded plaintext.

(The commands needed are very similar to those appearing in Exercise 3 of Computer Tutorial 9.)