PROGRAMMING AND DISCRETE MATHEMATICS (XX10190)
SEMESTER 2 MATHEMATICS: PROBLEM SHEET 4 – SOLUTIONS

All the large numbers in this sheet are assigned to MatLab variables in the MatLab script GKS4nums.m, to be found on Moodle.

1. Alice and Bob are unwisely using an encryption system that works by multiplication. That is, there is a chosen finite field $\mathbb{F}_q$ and Alice has chosen a secret element $a \in \mathbb{F}_q$. To encrypt a message m, which is also an element of $\mathbb{F}_q$, she simply computes am. Of course Bob cannot decrypt this, since he does not know a, so Alice and Bob use the Diffie-Hellman method to send m securely (only it isn't secure), using $\alpha(m) = am$ and $\beta(m) = bm$, where b is a secret element of $\mathbb{F}_q$ chosen by Bob.
Suppose that $q = 101$, which is prime, and Eve (who knows what the system is) hears Alice send 99 to Bob, then hears Bob send 31 to Alice and finally hears Alice send 60 to Bob. Find Alice's key, Bob's key and the message.
Eve knows ma, mab and mb so she computes $(ma)(mb)(mab)^{-1} = m$ and then $(ma)m^{-1} = a$ and $(mb)m^{-1} = b$. In the example given we have $ma = 99$, $mab = 31$ and $mb = 60$, so $m = 99 \times 60 \times (31)^{-1}$. So we need an inverse of 31 mod 101. Euclid's algorithm does this quickly (the inverse is -13) but here is an even quicker way: $31 = 132 = 11 \times 12$ so $m = 99 \times 60 \times 11^{-1} \times 12^{-1} = 9 \times 5 = 45$. Moreover, $a = 99 \times 45^{-1} = 11 \times 5^{-1} = 11 \times -20 = -220 = 83$, and $b = 60 \times 45^{-1} = 4 \times 3^{-1} = 1 + 3^{-1} = 35$. So Alice's key is 83, Bob's is 35, and Alice's secret message to Bob read "45".

2. After that experience, Alice and Bob switch to a system where the operation is exponentiation. Alice chooses a private key a (an integer) and Bob chooses a private key b. They agree on a cyclic group of prime order q generated by g. Alice writes down $m = g^r$ and computes $m^a = g^{ra}$; Bob sends back $(g^{ra})^b = g^{rab}$; Alice (who knows a^{-1} mod q) sends him $(g^{rab})^{a^{-1}} = g^{raba^{-1}} = g^{rb}$, and Bob computes $(g^{rb})^{b^{-1}} = g^{rbb^{-1}} = g^r = m$. This is better, because Eve sees only g^{ra}, g^{rb} and g^{rab}.
Suppose that $p = 47$ and that Alice and Bob work in the group $(\mathbb{F}_{47}^*)^2$ consisting of the set of squares of elements of $\mathbb{F}_{47}^*$, which is of order $q = 23$. Alice's private key is 9 and Bob's is 13. Alice's message to Bob is 6 (this is an allowable message because it is 10^2 mod 47). Work out exactly what is transmitted at each stage.
We first need to find the inverses of 9 and 13 mod 23. The inverse of 9 is $-5 = 18$ mod 23; the inverse of 13 is $-7 = 16$ mod 23. Then Alice sends $m^a = 6^9 = 3$ mod 47; Bob replies with $3^{13} = 36$ mod 47. Alice computes $36^{18} = 34$ mod 47 and sends that back to Bob, who computes $34^{16} = 6$ mod 47.

3. Not having understood what went wrong, Alice and Bob are again unwisely using an encryption system that works by multiplication as in Question 1. They have, though, learnt to use bigger numbers.

Repeat Question 1 with some big numbers. Suppose that $q = 10^{20} + 39$ (which is prime). Note that you will need to use the MatLab symbolic toolbox, since in regular MatLab $10^{20} + 39$ is indistinguishable from 10^{20}. Eve hears Alice send 71116957419887676546 to Bob, then hears Bob send 66775830479182144407 to Alice and finally hears Alice send 61081166643910584491 to Bob. Find Alice's key, Bob's key and the message.

Eve knows ma, mab and mb so she computes $(ma)(mb)(mab)^{-1} = m$ and then $(ma)m^{-1} = a$ and $(mb)m^{-1} = b$. In MatLab, this can be done as follows.

```
>> decodedm=mod(Alice2Bob1*Alice2Bob2/Bob2Alice,p)
decodedm =
1234567890123456789
>> AliceKey=mod(Alice2Bob1/decodedm,p)
AliceKey =
429537429537429537
>> BobKey=mod(Alice2Bob2/decodedm,p)
BobKey =
465435465435465435
```

4. Since that didn't work for them either, Alice and Bob now abandon multiplication for good and switch back to a system where the operation is exponentiation, as in Question 2. This time they don't bother about using only squares: they work in $\mathbb{F}_p^*$, the multiplicative group of the finite field $\mathbb{F}_p$ of prime order p. Alice sends Bob m^a; Bob sends back $(m^a)^b = m^{ab}$; Alice computes $(m^{ab})^{a'} = m^b$, where $aa' = 1 \pmod{p-1}$; and Bob can compute m. Again the chosen prime is $p = 10^{20} + 39$.

Assuming that for this question Alice's key is $a = 214768714768714769$ (the variable `q2a` in the worksheet), $b = 116358866358866359$ (`q2b`) and the message is 98765432109876543210 (`q2m`), what are the three messages exchanged, and how does Bob recover the message?

You have to use `myExptMod` *(see the sheet on the symbolic toolbox), or an equivalent. Then the following will do.*

```
>> q2A2B1=myExptMod(q2m,q2a,p)
q2A2B1 =
47188433175079902577
>> q2B2A=myExptMod(q2A2B1,q2b,p)
q2B2A =
```

```
2737045485992888096
>> q2arecip=mod(1/q2a,p-1)  [Note the p-1 here]
q2arecip =
7111795870492813123 7
>> q2A2B2=myExptMod(q2B2A,q2arecip,p)
q2A2B2 =
50158054049870769136
>> q2brecip=mod(1/q2b,p-1)  [Note the p-1 here]
q2brecip =
1970974454198463691
>> recovered=myExptMod(q2A2B2,q2brecip,p)
recovered =
98765432109876543210
```

GKS, 31/3/17