Methods of Cryptography in MATLAB

Jack Hodgkinson
1st Year Mathematics
Group 27

Question 1

For QUESTION 1, we were given numbers p, a and v for our group and told that we were Alice in the Diffie-Hellman key exchange, with p is the prime, g will be the smallest positive primitive root modulo p, a is our random number and v is the number sent to me by Bob. Part a) was to find g and part b) was to find the number u that you send to Bob and the shared key k.

My group was given the following values for p, a and v:

$$p = 694784159$$

$$a = 364387133$$

$$v = 514812135$$

Please find my solutions below.

a)

The task for part a) was to find g, which as stated in the question is the smallest positive primitive root modulo p. To find p, I decided the best method was to create a while loop in MATLAB, as it would iterate the loop until the given condition was satisfied. I have explained each line of my code with comments at the side.

```
>> p=sym('694784159');                %I have converted p to a symbolic number as
                                       p is a large integer,
                                       meaning we can carry out exact
                                       calculations with p.
>> g=2;                               %As 1 is not a primitive root, I have
                                       set the initial value of g to 2
>> while not(isPrimitiveRoot(g,p))   %This is the statement that the
                                       while loop is relying on, testing
                                       whether g is or not a primitive
                                       root modulo p
g=g+1;                                %If g is not a primitive root,
                                       increase g by 1
end                                   %End the loop once g is a primitive
                                       root modulo p, or the initial condition
```

```
                                        becomes false.
>> g                                   %Output the value of g
g =
11
```

The code outputs $g = 11$, therefore 11 is the smallest positive primitive root modulo 694784159.

b)

Then for part b, we were asked to find u that we send to Bob and the shared key k. With Diffie-Hellman Key Exchange, we are asked to calculate u, which is the remainder of dividing g^a by p. Thinking about this with regards to modular arithmetic, $g^a \equiv u \bmod p$. Also for Diffie-Hellman, Alice needs to compute the shared key k, which by definition is the remainder of dividing v^a (where v is what Alice receives from Bob) by p, or $v^a \equiv k \bmod p$.

As MATLAB stores values from previous commands, we do not need to remind MATLAB of the values of g and p as they are stored in the Workspace. I turned both a and v into symbolic numbers as they are large integers and to increase the accuracy of the calculations. I decided to use the powermod function in MATLAB to compute u and k as they are both congruent to exponents. You can see my MATLAB code below.

```
>> a=sym('364387133');
>> v=sym('514812135');
>> u=powermod(g,a,p)
u =
18122020
>> k=powermod(v,a,p)
k =
20201218
```

From the code above, it can be concluded that

$$11^{364387133} \equiv 18122020 \bmod 694784159$$

$$514812135^{364387133} \equiv 20201218 \bmod 694784159$$

hence

$$u = 18122020, v = 20201218$$

Question 2

For QUESTION 2, we were asked to a) carry out the Miller-Rabin test on a number n which we are given with base 7 and store the sequence of remainders in a vector. Also, we should print out r and the sequence of remainders. For part b) we should use the sequence of remainders to find the prime factorisation for n. I have answered each part of the question sepearately below.

a)

For a), we were asked to carry out the Miller-Rabin test on n with base 7.

The Miller-Rabin test is a primality test to test whether a number n is composite or not. As the test is inconclusive, if n passes the test it is said to be 'probable prime' but if n fails the test it is said to be 'definitely composite.' To carry out the test, we take $n-1$ and write it in the form $2^r s$, where r is a positive integer and s is an odd integer. We take an integer a which is coprime to n as the base, which we have been given that the base is 7 therefore $a = 7$ in the context of the question. Then we must calculate the remainders of $a^s, a^{2s}, a^{4s} ... a^{2^r s}$ on division by n, where $2^r s{=}n-1$. After outputting the remainders in a sequence, there are a few conditions that need to be satisfied to pass the Miller-Rabin test:

- The last term in the sequence is 1
- The first term in the sequence is 1 OR the first 1 in the sequence is preeceded by $n-1$

I have carried out the Miller-Rabin test using the values we were given:

$$n = 94315998521858388311455322520560147698349301902342414918209$$

$$a = 7$$

I have explained my code on each line below:

```
>> clear all  %clear the workspace so no values from the previous
              questions are used
n=sym('94315998521858388311455322520560147698349301902342414918209');
%turned n into a symbolic number as n is a very large integer and to
```

```
improve accuracy of further calculations.
s=sym(n-1);          %As we need to find r and s, I have done this through
r=0;                  a while loop. I have set initial values of r=0 and s=n-1.
while mod(s,2)==0    %The while loop will keep dividing s by 2 until there
s=s/2;               is a remainder (i.e mod(s,2)=1). For each time s is
r=r+1;               divided by 2, r will increase by 1 to reflect how many
end                  powers of 2 (n-1) is divisible by.
>> s                 %Output the value of s

s =

1473687476904037317366489414383752307786707842224100233097

>> r                 %Output the value of r

r =

6
```

Hence we have found values for r, s such that $n-1 = 2^r s$

$$r = 6$$

$$s = 1473687476904037317366489414383752307786707842224100233097$$

Now we need to compute the sequence of remainders from a^{2s} to $a^{2^6 s}$ on division by n. We know that $a = 7$, so I have added this line into my code. I have created a new vector k to reflect the powers of 2, as we need to calculate remainders from $a^{2s} = a^{2^0 s}$ to $a^{2^6 s}$, hence k is the vector containing integers from 0 to 6. I have created a second vector q which is the sequence of the remainders. Each value of q is the remainder dividing $a^{2^r s}$ by n, i.e $q \equiv a^{2^r s} \bmod n$. From the sequence of remainders we will be able to conclude if n passes the Miller-Rabin test. I have explained this in my code below.

```
>> a=7;                             %the base we are given in the question
k=[0:r];                            %create a vector k which is the powers of 2
                                    needed to calculate the remainders.
q=powermod(a,sym(2.^k) *sym(s),n);  %to produce remainders for each element
                                     in k, we use the operator 2.^k to do 2 to
```

```
                                         the power of each element of k. I have made
                                         s and 2.^k symbolic numbers as they are very
                                         large integers and to improve accuracy of
                                         the calculations. To find the remainders,
                                         I have used powermod which finds the
                                         remainder from division of an exponent
                                         (in this case 2^r*s) by n. This will also
                                         create q as a vector.
>> q=q'                                 %The ' turns q from a row vector to a column
                                        vector. I have done this so you can see the
q =                                     elements of q clearer.

4964895189558091513662837052010960713621203414408739192625
77816224078295983330833604397149647100239267257701424624977
8347129248182165805257920262680326741207618127019901420038
94315998521858388300160088078491537496190205055368508479001
1
1
1
```

From the vector q above, we can see that the last term is 1. However, the first time in the sequence is not 1 and the first 1 in the sequence is not preceeded by $n-1$. Therefore we can conclude that n **fails** the Miller-Rabin test and so n is **definitely composite**.

b)

For part b), the question asks us to take the vector of remainders from a) (my q vector), add 1 to them and calculate the greatest common divisor of $q(k)+1$ and n, denoted by $\gcd(q(k)+1, n)$, where $g(k)$ is the $k-th$ element of the vector q. Then we are to subtract 1 from each element of q and calcuate $\gcd(q(k)-1, n)$. Then to use the numbers to calculate the prime factorisation for n.

I did this through implementing a code in MATLAB. I first of all created a new vector u which is every element from q but with 1 added. I then created the vector $gcdu$ which is the $\gcd(u, n)$ and displayed all elements. I also created a vector v which is all elements of q subtracted by 1. Then created *gcdv* which is $\gcd(v, n)$ and dispalyed all elements. Please see my code below.

```
>> u=q+1;                    %Create the vector u=q+1
gcdu=gcd(u,n)                %Create vector gcdu which is gcd(u,n) for all
                             elements of u.
gcdu =

50100420140328852673
25050210070164426337
1
75150630210493279009
1
1
1

>> v=q-1;                    %Create the vector v=q-1
gcdv=gcd(v,n)                %Create vector gcdv which is gcd(v,n) for all
                             elements of v.
gcdv =

1
50100420140328852673
125502604911873446690656631633043404 8801
125502604911873446690656631633043404 8801
9431599852185838831145532252056014769834930190234241 4918209
9431599852185838831145532252056014769834930190234241 4918209
9431599852185838831145532252056014769834930190234241 4918209
```

As we are looking for the prime factorisation of n, the next step I took was to find out which of the elements of both *gcdu* and *gcdv* are prime. I did this using the *isprime* function in MATLAB, to easily discard all of the composite numbers. All of the prime numbers with this function will return a 1, and all composite numbers will return a 0. Please see my MATLAB code below:

```
>> isprime(gcdu)      %tests all values of gcdu to see if they are prime
```

```
ans =

7×1 logical array

1
1
0
1
0
0
0

>> isprime(gcdv)       %tests all values of gcdv to see if they are prime

ans =

7×1 logical array

0
1
0
0
0
0
0
```

As we can see from the results above, $gcdu(1), gcdu(2), gcdu(4)$ and $gcdv(2)$ are all prime numbers - where $gcdu(k)$ denotes the $k-th$ element of the vector $gcdu$. I then created a vector z which included all of these prime numbers to see if we had any repetition. Please see my code below.

```
>> z=[gcdu(1),gcdu(2),gcdu(4),gcdv(2)];
>> z'

ans =

50100420140328852673
25050210070164426337
```

```
75150630210493279009
50100420140328852673
```

As we can see from the displayed z vector above, elements $z(1)$ and $z(4)$ are the same so we do not need to repeat this number in our prime factorisation. I then decided to run a test on the vector z to see if all elements of z multiplied by each other (except for repeating numbers) were equal to n. I did this using an if statement in MATLAB. I created a new variable called $primefac$ which is equal to $z(1) \times z(2) \times z(3)$. Then I asked MATLAB that if these three numbers multiplied are equal to n then to confirm that is the prime factorisation. Please see my code below:

```
>> primefac=z(1)*z(2)*z(3);   %created new variable primefac
>> if primefac==n             %Asks MATLAB if primefac=n, then display
                              the prime factorisation.
disp('z(1)*z(2)*z(3) is the prime factorisation for n')
end
z(1)*z(2)*z(3) is the prime factorisation for n
```

Hence we can conclude that the prime factorisation for n is $z(1) \times z(2) \times z(3)$.

Therefore.

$n = 50100420140328852673 \times 25050210070164426337 \times 75150630210493279009$

Question 3

For Question 3, we were asked to see whether the signature (r, s) is valid for H. This is referring to the Digital Signature Algorithm, where H is known as the hash of a document D. Normally, we would have to calculate p, q, r, s, g, y but luckily we are given them in the question. You can see the values for each of these variables in my code. The question also wanted us to calculate values for u_1 and u_2, which solve the congruences $su_1 \equiv H \bmod q$ and $su_2 \equiv r \bmod q$. I will first calculate u_1 and u_2 and then explain the process to verify the signature. I have written $u1$ for u_1 and $u2$ for u_2 in my code.

```
>> clear all                          %clear the workspace so no values
```

```
                                    from the previous questions are used
p=sym('30167674936870980426367');
q=sym('17456345243');
g=sym('18008617784390347685963');   %let p,q,g,y,H,r,s all be converted to
y=sym('61726472517312324125430');    symbolic numbers to improve the accuracy
H=sym('600517321');                 of the calculations
r=sym('2143242993');
s=sym('436861398');

[a,b,c]=gcd(s,q);         %This function calculates values for a,b and c from
                          the gcd(s,q). a = the greatest common divisor of s
a =                       and q, b and c are solutions that solve the linear
                          Diophantine equation sb+qc=a. As the gcd(s,q)=1, it
1                          can be said that sb+qc=1 and by definition b is the
                           multiplicative inverse.
b=


-7498490397

c =

187656749

>> u1=mod(sym(b*H),q)     %We can use the inverse b to solve the two
                          congruences su1. By defintion of the multiplicative
u1 =                      inverse,we can multiply both H and r by the
                          inverse b to and then use congruences to get the
10295297556               values for u1 and u2.

>> u2=mod(sym(b*r),q)

u2 =

14676908269
```

Therefore we have worked out that $u_1 = 10295297556$ and $u_2 = 14676908269$.

To verify the signature, we need to calculate the remainder on dividing $g^{u_1}y^{u_2}$ by p. In other words, we need to find a remainder to the congruence $g^{u_1}y^{u_2} \equiv x \bmod p$, where we are denoting x to be the remainder. By modular arithmetic, we can split up the congruence to be $g^{u_1} \equiv r_1 \bmod p$ and $y^{u_2} \equiv r_2 \bmod p$, with r_1, r_2 being the remainders from those congruences. Then $r_1r_2 \equiv x \bmod p$. By DSA, if $r \equiv x \bmod q$ then the signature is accepted. In my code, I have used r_1 and r_2 as stated in this paragraph, but let $r_3 = x$ and r_4 be the result of $r_3 \bmod q$. I have also added an if loop to test whether the r_4 value I calculated is equal to the r value we are given. If they are equal, the code will display 'The signature is valid.' I will explain my code below.

```
>> r1=powermod(g,u1,p);        %r1 is the remainder of dividing g^u1 by p,
                               hence why we are using powermod
r2=powermod(y,u2,p);           %r2 is the remainder of dividing y^u2 by p,
                               hence why we are using powermod
r3=mod(r1*r2,p);               %r3 is the remainder of dividing r1r2 by p.
>> r4=mod(r3,q)               %r4 is the remainder from dividing r3 by q,

r4 =

2143242993

>> if r==r4                       %If our r4 value matches the r value given
disp('The signature is valid.')   to us, then the signature is valid, if not
else                              the code will tell us signature is not valid.
disp('The signature is not valid.')
end
The signature is valid.           $as r4=r, the signature is valid
```

Hence from the code above, we can conclude that (r, s) is a valid signature for H.

Question 4

For Question 4, we were asked to use our student number to a) find the smallest prime p such that $p > 10S^4$ and $(p-1)/2$ is also prime and to also find the largest prime q such that $q < 4S^5$ and $(q-1)/2$ is also prime. Then using our p and q in RSA with the public key $e = 65537$, to calculate the private key d. For b) we were asked to decrypt the messages given to us using our private key, and to divide the numbers into 2 digit blocks and to use the table given to us to figure out the text (see *Figure 1*). I will do each part of the question seperately below.

a)

We are tasked with calculating p, q which satisfy the conditions stated above and then to use e to calculate the private key d. My student number is S, so let $S = 10727774$. To find p, I created a while loop which starts at the first prime greater than $10S^4$ as the initial value and will search through consecutively larger primes until a prime is found where $(p-1)/2$ is also prime. To find q, I created a second while loop which starts at the largest prime less than $4S^5$ as the initial value and will search through each consecutively smaller primed until a prime is found where $(q-1)/2$ is also prime. Please see my code below for further explanation.

```
>> clear all                     %Clears workspace so no previous values used.
S=sym('10727774');               %S is my student number, sym(S) to make large
                                 %integer calculations more accurate.
p=nextprime(10*S^4);             %We set the initial value of p to be the next
                                  prime which is greater than 10*S^4.
while not(isprime((p-1)/2))      %Introducing a while loop which will test
                                  increasing primes p > 10*S^4 to see if (p-1)/2 is
                                 also prime. Once we reach a value for p where
p=nextprime(p+1);                (p-1)/2 is also prime the loop will stop iterating.
end
q=prevprime(4*S^5);              %We set the initial value of q to be the largest
                                 prime which is less than 4*S^5.
while not(isprime((q-1)/2))      %Introducing a while loop which will test
                                 decreasing primes q < 4*S^5 to see if (p-1)/2 is
                                 also prime. Once we reach a value for q where
q=prevprime(q-1);                (q-1)/2 is also prime the loop will stop iterating.
end
p                                %Output the value of p
q                                %Output the value of q
```

```
p =

1324458829683486659054424 70403

q =

568339799885957456414036822741525723
```

Hence we have our personal primes for RSA,

$$p = 13244588296834866590544247 0403$$

$$q = 568339799885957456414036822741525723$$

To find the private key, we need to find a value d that satisfies the congruence $de \equiv 1 \bmod (p-1)(q-1)$. As explained in Q3, we can use multiplicative inverses to find the value of d. As $\gcd(65537, (p-1)(q-1)) = 1$, we can find the value x (multiplicative inverse of $de \equiv 1 \bmod (p-1)(q-1)$) and multiply both sides of the congruence by y so we have $dex \equiv x \bmod (p-1)(q-1)$. By using the fact that $xe \equiv 1 \bmod (p-1)(q-1)$, we can simplify the congruence to $d \equiv x \bmod (p-1)(q-1)$. As congruences are symmetric, we can change this to $x \equiv d \bmod (p-1)(q-1)$ and solve this using the mod function in MATLAB which will generate the value of d which we require. In my code, I have used t to denote $(p-1)(q-1)$.

```
>> e=sym('65537');      %We are given the public key e, but transforming
                        it to a symbolic to improve the accuracy of large
                        integer calculations.
t=sym(p-1)*sym(q-1);    %t is symbolic (p-1)(q-1) again to improve accuracy
[gcd,x,~]=gcd(e,t);     %With this function, we are calculating the gcd(e,t)
                        and calculating the x value which will solve the linear
gcd =                   Diophantine equation ex-ty=gcd(e,t). As the gcd=1,
                        by definition x is a multiplicative inverse of the
1                       congruence. We will use the x value to calculate d.


x =

-319534018557861233886523631147366334846126955738075674099469283 67
```

```
>> d=mod(x,t)            %As explained above, we have rearranged the congruence
                         to calculate d from the values x and t that we are
d =                      given. We do this using the mod functiom.

43320864766164098341114348655590280558559922320535586628359751877
```

The private key for RSA is

$d = 43320864766164098341114348655590280558559922320535586628359751877$

b)

We were now asked to use our private key d to decrypt two messages which were assigned to each student number. Then we were told to divide the decrypted message into 2 digit blocks, and convert each 2 digit block to a corresponding letter in *Figure 1*, where 00 denotes a space, and then join the two messages. The two encrypted messages I was given I have called $c1$ and $c2$ in my code and are:

$c1 = 595193670073209661864930803864278463005166580546990297345192863 04$

$c2 = 454817485058843334566305552052400742200801935480495257108369521 91.$

In order to decrypt the messages through RSA, we need to calculate the remainder on dividing c^d by N, where $N = pq$ is known as the modulus. If we call the remainder m to denote the message, then we can represent this using the congruence $m \equiv c^d \bmod N$. I have shown this in my code below.

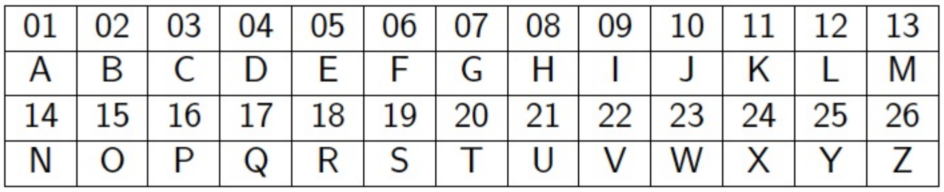
```
N=p*q;     %calculating the modulus N by p*q
c1=sym('59519367007320966186493080386427846300516658054699029734519286304');
c2=sym('45481748505884333456630555205240074220080193548049525710836952191');
%as c1 and c2 are large numbers, I have made them symbolic numbers to improve
the accuracy of the calculations.
>> m1=powermod(c1,d,N)  %as c1^d is an exponent, use powermod to calculate m1
m2=powermod(c2,d,N)     %as c2^d is an exponent, use powermod to calculate m2

m1 =
1301200805130120090319000919000315140305181405040023092008000112


m2 =

1200161519190902120500231518120419000400130001181319201815140700
```

Now that we have our decrypted messages $m1$ and $m2$, we need to do as the question asks and split them up into 2 digit blocks and use *Figure 1* to unlock the message.



01	02	03	04	05	06	07	08	09	10	11	12	13
A	B	C	D	E	F	G	H	I	J	K	L	M
14	15	16	17	18	19	20	21	22	23	24	25	26
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Figure 1: This table gives a letter to each corresponding 2 digit block

I will split up each message below into 2 digit blocks and use each block to find its corresponding letter. For example, the first 2 digit block is 13 which corresponds to the letter M in the table, so on the line below I will write M. I will do this for each letter until I have a message, and take 00 to be a space.

$m1$ = 13|01|20|08|05|13|01|20|09|03|19|00|09|19|00|03|15|14|03|05|18|14|05|04|00|23|

MATHEMATICS IS CONCERNED W

|09|20|08|00|01|12|

ITH AL

$m2$ = 12|00|16|15|19|19|09|02|12|05|00|23|15|18|12|04|19|00|04|00|13|00|01|18|13|19|

L POSSIBLE WORLDS D M ARMS

|20|18|15|14|07|00|

TRONG

Connecting all of the text together, we have decrypted the messsage:

MATHEMATICS IS CONCERNED WITH ALL POSSIBLE WORLDS D M ARMSTRONG

Which as we know, is the famous quote by David Malet Armstrong.