CAB340 Assessment 3
Public key encryption

1 Number theory

a. Bézout's identity states that for positive integers a and b, there always exist integers m and n such that

$$am + bn = \gcd(a, b).$$

Also, recall that $a \equiv b \pmod{n}$ means that there exists an integer ℓ such that

$$a - b = \ell n.$$

This is called *modular equivalence* or *modular congruence*.

Recall that (if it exists) the multiplicative inverse of p modulo q, is defined to be the unique number p^{-1} such that $p^{-1}p \equiv 1 \pmod{q}$.

i. Apply the definition of modular equivalence and write down what $p^{-1}p \equiv 1 \pmod{q}$ means. **1 mark.**

ii. Rearrange what you get and apply Bézout's identity to conclude that if $\gcd(p, q) = 1$ then p^{-1} exists (modulo q). **1 mark**.

b. The *Division algorithm* tells us that, given positive numbers a and b, with $a \geq b$, there always exist integers q and r, with $0 \leq r < b$ such that

$$a = bq + r.$$

(think of primary school division before you learned about decimals) q is called the *quotient* and r is called the *remainder*. It can be shown that

$$\gcd(a, b) = \gcd(b, r).$$

The *Euclidean algorithm* makes use of this fact to calculate $\gcd(a, b)$ by repeatedly replacing the larger number by a remainder (which is always smaller). The algorithm is:

i. Set $a_0 = a$, $b_0 = b, j = 0$

ii. Use the division algorithm to find q_j and r_j with $0 \leq r_j < b_j$ such that

$$a_j = b_j q_j + r_j$$

iii. If $r_j > 0$ then set $a_{j+1} = b_j$, $b_{j+1} = r_j$, increment j and go back to the previous step

iv. Output r_{j-1} as $\gcd(a, b)$.

We can keep track of all values in a table. For example, we can calculate $\gcd(27, 16)$ by:

j	a_j	b_j	q_j	r_j	
0	27	16	1	11	$(27 = 16(1) + 11)$
1	16	11	1	5	$(16 = 11(1) + 5)$
2	11	5	2	1	$(11 = 5(2) + 1)$
3	5	1	5	0	$(5 = 1(5) + 0)$

and we find $\gcd(27, 16) = 1$ from the second last value of r_j. The final column is included for illustration and is not necessary.

Use the Euclidean algorithm to calculate $\gcd(31, 19)$. Organise your work in a table as in the example above. **1 mark**.

c. The Euclidean algorithm can be extended to find m and n in Bézout's identity. Suppose that we have a_j, b_j, q_j, r_j for $j = 0, \ldots, \ell, \ell+1$ from the Euclidean algorithm. Then we have

$$\gcd(a, b) = r_\ell.$$

We also have, with some rearrangement,

$$a_\ell - b_\ell q_\ell = r_\ell.$$

Here a_ℓ and b_ℓ are defined in terms of $a_{\ell-1}$ and $b_{\ell-1}$, so we can substitute in these definitions and find an equation relating $a_{\ell-1}$, $b_{\ell-1}$ and r_ℓ. Repeatedly substituting in, we eventually find an equation relating a, b and r_ℓ.

This leads to the *Extended Euclidean Algorithm*:

- **i.** Run the Euclidean algorithm to obtain ℓ and values for q_j.
- **ii.** Set $j = \ell - 1$, $m_\ell = 1$, $n_\ell = -q_\ell$
- **iii.** Set $m_j = n_{j+1}$, $n_j = m_{j+1} - n_{j+1} q_j$
- **iv.** If $j > 0$ then repeat decrement j and go back to the previous step
- **v.** Output m_0 and n_0.

We can keep track of the values in a table, filling in the necessary values for q_j from the table for the Euclidean algorithm. For example, with $a = 27$, $b = 16$ we obtain:

j	q_j	m_j	n_j	
2	2	1	-2	$(11(1) + 5(-2) = 1)$
1	1	-2	3	$(16(-2) + 11(3) = 1)$
0	1	3	-5	$(27(3) + 16(-5) = 1)$

The last column is for illustration only, with a_j and b_j obtained from the table for the Euclidean algorithm.

Run the extended Euclidean algorithm with $a = 31$, $b = 19$ using the values of q_j obtained from your previous run of the Euclidean algorithm. Organise your results in a table as above. **1 mark.**

d. Write out Bézout's identity for $a = 31$, $b = 19$ with all values, including m and n filled in. **1 mark.**

2 RSA encryption

For the following questions, we are using the RSA cryptosystem. For each of the questions below, write down **the operation you need to perform** in addition to **the answer that you get**. Wolfram Alpha is suitable for all the calculations you will need to do.

a. Given the primes $p = 2027$ and $q = 2593$, and using $e = 7$

- **i.** Generate the corresponding public RSA key. **1 mark.**
- **ii.** Generate the corresponding private RSA key. **1 mark.**

b. Using the public key above, encrypt the message 1024. **1 mark.**

c. Using the private key above, decrypt the message 3054908. **1 mark.**

d. You are given the public key $(n = 7354943, e = 7)$. Determine the private key by first factoring the public key. Wolfram Alpha is able to perform the necessary factorisation. **1 mark.**

3 Digital signatures

a. Digital signatures are sometimes used for authenticating users in network protocols. SSH and TLS, for example, can both use such a mechanism. Consider the following protocol:

- The server sends Alice a randomly chosen number.
- Alice digitally signs the number and sends the signature back to the server.
- The server checks the signature using Alice's public key. If the signature is valid, then the server accepts Alice's connection request.

Suppose that Alice uses the same private key to log in to a server and sign her emails. Show how a server could forge emails from Alice. For this exercise, assume that the authentication mechanism signs the bare challenge without hashing, while for emails Alice signs the hash of the message. **2 marks.**.

b. Suggest a modification to the authentication protocol which would defeat this attack. **2 marks.**

4 Diffie-Hellman key agreement

For this question we will be considering the original Diffie-Hellman key agreement algorithm, which is like the authenticated version, but excludes the signatures and verifications.

a. For Diffie-Hellman we need to choose a public *generator* g modulo p (where p is prime). If g is a generator, then for every x with $\gcd(p, x) = 1$, there must exist a k such that

$$g^k \equiv x \pmod{p}.$$

This means that we can take discrete logarithms with base g for any x which is not a multiple of p.

We can test if g is a generator by finding its *order*, which is the smallest k such that $g^k \equiv 1 \pmod{p}$. g is a generator modulo p if and only if the order of g is $p - 1$.

Using $p = 2027$, find all generators between 1 and 10. You can use Wolfram Alpha to find the order of g modulo p by typing, for example, `order of 2 modulo 2027`, substituting in your value of g for 2. **1 mark.**

b. Using $p = 2027$ and g the smallest generator that you found in the previous question, suppose that Alice and Bob perform Diffie-Hellman key agreement, where Alice's secret is 424 and Bob's secret is 1746. What messages do Alice and Bob send to each other? **1 mark.**

c. What is the key that Alice and Bob agree on? **1 mark.**

5 Secret sharing

Consider the following secret sharing scheme for m parties for sharing a secret n-bit string x.

- The dealer generates $m - 1$ random n-bit strings $r_1 \dots r_{m-1}$.

- The dealer sends r_j to party j for $j = 1 \ldots m-1$.
- The dealer sends $r_1 \oplus \cdots \oplus r_{m-1} \oplus x$ to party m

a. Explain how all of the parties together can reconstruct the secret x. **1 mark.**

b. Explain why any $m-1$ parties do not have enough information to reconstruct the message. It may be helpful to note that for $m = 2$ this scheme corresponds to the one-time-pad. You may also, for the purposes of your explanation, consider a single bit secret (i.e. $n = 1$). **1 mark.**

c. Suppose that 3 parties share two secrets x and y using the above scheme. Their shares are s_j for x and t_j for y. Show that if each party forms $u_j = s_j \oplus t_j$ and then they get together to reconstruct the secret from the u_j's, the result is $x \oplus y$. **1 mark.**