Cryptology - Week 4 worksheet

These exercises are to consolidate the lecture material from week 4. This worksheet is on RSA and ElGamal encryption and is an introduction to using SageMath.

You will need to use SageMath www.sagemath.org, which is Python-based, for this sheet and future sheets, and during the midterm and coursework if you are taking this course as a major. SageMath has a lot of support for abstract algebra. Useful resources are the "quickrefs" for number theory[1] and abstract algebra[2].

SageMath is open source and you can download it on your own machine, and/or you can use it on the Linux lab machines in MVB 2.11, MVB 1.15 or QB 1.80. On Linux lab machines, log in using your university credentials, open a terminal, then type the following to run SageMath:

```
module load sagemath
sage
```

You will need to do this (both instructions) every time you open a new terminal.

- Questions 1-4 are to do by hand.
- For questions 5-7, we recommend using SageMath.
- A version of question 3 is an option for an exam question.

1. (a) Let φ be the Euler φ-function. Prove that:
   (i) If p is prime, the $\varphi(p) = p - 1$.
   (ii) If p and q are distinct primes, then $\varphi(pq) = (p-1)(q-1)$.
   (iii) If p is prime, then $\varphi(p^2) = p(p-1)$.

   (b) Which elements should you remove from $G = \mathbb{Z}/pq\mathbb{Z}$ in order for $(G, * = \times \pmod{pq})$ to be a group? What is the resulting size of G?

[1] https://wiki.sagemath.org/quickref?action=AttachFiledo=viewtarget=quickref-nt.pdf
[2] https://wiki.sagemath.org/quickref?action=AttachFiledo=viewtarget=quickref-algebra.pdf

2. (a) Using double-and-add, compute $69 \cdot 73 \pmod{1000}$. Write out your steps and compute the number of additions required.
Hint: the binary expansion of 69 is 1000101.

   (b) How would you efficiently compute $2047 \cdot 7879$?

3. **This is an option for an exam question**

   (a) Hellman contacts you to tell you that he wants to send you an encrypted message using ElGamal encryption. You choose parameters $p = 37$, $g = 2$, and $sk_A = 7$, and send your public key to Hellman (you do not have to compute this now). Hellman replies with the ciphertext

$$(pk_H, enc_m) = (9, 13).$$

   (i) Using square-and-multiply, compute your shared secret key with Hellman.

   (ii) Decrypt the message.
(Note: the 'message' is just a number mod 37).

   (b) You ask Hellman to share the message with Bob. You observe Hellman sending the ciphertext

$$(pk_H, enc_m) = (9, 8)$$

   to Bob. Compute Bob and Hellman's shared secret.

4. (a) I'd like to make RSA super efficient and reduce the burden of exponentiation by setting $e = 1$. Why is this a terrible idea?

   (b) Ok, $e = 1$ is perhaps a bit too lax. But why not $e = 2$? More generally, why do I need the Euler-phi function condition on e?

5. Modular arithmetic on a computer is very easy, using the operator %. If *all* operations are happening $\pmod{m}$, then constantly typing % is a bit tedious. We can set up a mathematical space so that we're always working modulo m, by using `Zmod(m)` or `IntegerModRing(n)`. To familiarize yourself with this, try some of the questions from the first workshop of term using this way of writing modular arithmetic. Then:

   (a) By checking all the possible values of $2^a \pmod{31}$, show that 2 does not generate $\mathbb{F}_{31}^*$ as a multiplicative group.

   (b) Find a generator of $\mathbb{F}_{31}^*$ as a multiplicative group.

   (c)* How many possible choices of generator are there for $\mathbb{F}_{31}^*$?

6. This question is a toy example of RSA.Set $p = 307$, $q = 311$, and $n = p \cdot q$. Note that p and q are prime numbers.

   (a) Compute the RSA secret key corresponding to the RSA public key $(247, n)$.

(b) The values $m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7$ below are a message that has been encrypted using the public RSA key $(247, n)$. Decrypt this message and translate it into plaintext by assigning the value 00 to A, 01 to B, etc., up to 25 to Z, and assigning the value 26 to !.

$$m_0 = 94755$$
$$m_1 = 87565$$
$$m_2 = 41862$$
$$m_3 = 49231$$
$$m_4 = 34234$$
$$m_5 = 17479$$
$$m_6 = 26771$$
$$m_7 = 87503.$$

(c) Give 2 examples of an invalid public key. Justify your answer.

(d) (Optional) Write a function called `RSA(p,q,m)` that takes as input primes p, q and a message m, and outputs a valid public key, secret key and ciphertext. Write another function that decrypts a ciphertext (with given public key and secret key).

7. (Optional) Write a function that performs square-and-multiply.