Assignment # 5

Homework

Homework problems are a preparation for the quizzes. They are *not* graded. Please use the `mywpi` forum to post questions you have on these problems.

- 8.1, 8.2, 8.3, 8.5, 8.8, 8.9, 8.12, 8.13, 8.14, 8.18

Project

Note: For submissions on `mywpi`: Please submit a single pdf file containing your results. Please submit source code as a separate file, but make sure to have it listed in the pdf as well.

1. 8.4 (if you use sage for part of the problem, please provide the code as well)

2. Implement the square and multiply algorithm using a computer language of your choice. The program should print all intermediate results. For sage and Python, you can use the following template:

   ```
   def my_pow(b,e,m):
       """ Computes b^e mod m using the square and multiply algorithm"""
       x = pow(b,e,m) # remove this line and place your code here instead
   return x
   ```

   Compute the following exponentiations $a^e \bmod p$ using your program:

   (a) $a = 235973, e = 456789, p = 583903$

   (b) $a = 984327457683, e = 2153489582, p = 994348472629$

   Print the output of your program (including the intermediate steps) and turn it in together with your source code.

3. The goal of this problem is to implement the Diffie Hellman Key Exchange.

   (a) Choose a pseudo random generator and initialize it with a $seed = 12345$ (e.g. `set_random_seed()` in sage). You can use a PRNG of your choice. Please comment on the security of your scheme given that you will use the same PRNG to generate the secret DHKE parameters A, B.

   (b) Next, we have to generate a large prime as modulus for our cyclic group. Please use a tool of your choice to generate a prime p of exactly 1024 bits. Does this prime have to be random? Justify your answer.

(c) In order to increase the security of the scheme we use a *safe prime*. A safe prime p_s is of the form $p_s = 2 \cdot p + 1$, where p is also prime. The group we will use for DHKE in this problem will be multiplicative group $\mathbb{Z}_{p_s}^*$. One way of choosing the DHKE parameter g is choosing it as a random element from $\mathbb{Z}_{p_s}^*$. Explain why using a safe prime helps ensuring that finding an element g of high order q is easy. Make your DHKE program choose g at random from $\mathbb{Z}_{p_s}^*$. The safe prime used should also have 1024 bits. If you have trouble generating one yourself, please use: $p_s =$

```
13649309113364983616428936333868200294402937901407703313060436392225659 5\
```

(d) Next, generate two random parameters A and B. Write your code so that A, B can only take valid parameters.

(e) Since the *Decisional Diffie Hellman* problem is easy to solve for the given group, please apply SHA-224 to the computed $k_A = k_B$ to get a symmetric-sized key from the scheme. Hash functions are a convenient way of turning a random group element into a random bit sequence. Please use big endian conversion to represent your integer as SHA-224 input (e.g. `hex(k_A).decode('hex')` in Python 2 or sage).

4. The ElGamal encryption scheme is non-deterministic: A given message m has many valid encryptions.

(a) Why is the ElGamal encryption scheme non-deterministic?

(b) How many valid ciphertexts exist for each message m (general expression)? How many are there for the system in problem 8.13 from the book (numerical answer)?

(c) Consider the case that for two messages $m_1 \neq m_2$ the same session parameter $y_1 = y_2$ has been chosen for the ElGamal encryption. This kind of behavior occurs if no or a bad cryptographic PRNG is being used. Show how the message m_2 can be recovered from a known message ciphertext pair $\langle m_1, c_1 \rangle$ if the same y is used.

Good Luck and Have Fun!