Problem Set 5

Due: March 6th, 5pm

1 Formal proofs

For each problem in this section, you must give a complete and rigorous proof. Your arguments should be clear, logically ordered, and written in full sentences.

In particular, state all assumptions explicitly and define all notation. Justify every nontrivial step.

1. (2 points) Let p, q be prime numbers and $n = pq$. Suppose (n, e, d) is a valid RSA key triple. Prove that for *any* $M \in \mathbb{Z}_n$, $M^{ed} \equiv M \pmod{n}$.

2 Demonstrations

For problems in this section, you still need to give complete mathematical reasoning to support your answers. I should be able to follow and understand your work, but it doesn't have to be organized into a formal proof.

2. (2 points) Consider RSA with primes $p = 11$ and $q = 17$.
   (a) Compute n and $\phi(n)$ and verify that $e = 7$ is a valid encryption exponent. Then compute the corresponding decryption exponent d.
   (b) Use your public key (n, e) to encrypt the message $M = 5$.
   (c) Find the private key (n, d). Use this to decrypt the ciphertext from part (c) and verify you recover $M = 5$.
   (d) Code-breaking: Suppose an adversary intercepts a different RSA communication with public key $(n, e) = (221, 5)$ and ciphertext $C = 144$. Find the original plaintext M.

   Do all calculations by hand! Problems like this are excellent practice for exams, where you won't have access to computational tools. Show every step of your arithmetic clearly.

3. (1 point) Suppose n is a positive integer that is a product of two distinct primes p, q. Write down a formula to compute p, q in terms of $n, \phi(n)$.

4. (1 point) Consider the following naive algorithm for factoring a positive integer $n > 1$:

```
for i = 2, 3, 4, ..., floor(sqrt(n)):
    if n % i == 0:
        return i
return n  # n is prime
```

   (a) Explain why this algorithm correctly finds the smallest prime factor of n (or determines that n is prime).

   (b) Show that the number of steps (iterations of the loop) required by this algorithm is $O(\sqrt{n})$.

   (c) Show that number of steps (iterations of the loop) required by this algorithm is not $O(n^a)$ for any $a < 1/2$. Conclude that it is not $O(\log n)$.

3 Explorations

In this section, I'm looking for answers that are supported by evidence, but you don't have to prove your answers.

5. (1 point) Write a computer program to implement the RSA encryption and decryption process. Use it to generate a public/private key pair, encrypt a message, and then decrypt it. Show your code and the results of your encryption and decryption.

6. (2 points) Implement the binary exponentiation algorithm in a programming language of your choice. Use it to compute $3^{123456789} \bmod 1000000007$. Provide a screenshot of your code.

7. (1 point) Two spies, Alice and Bob, need to communicate the location of a secret document hidden in a book at the Eisenhower Library. To avoid detection, Alice encrypts the book's title using RSA and sends the ciphertext over an insecure channel. Your mission, should you choose to accept it, is to crack their encryption and discover where the secret lies.

   Alice uses RSA with public key (n, e):

   ```
   n = 1156155289961087756632600159458853982244325906947295006670251
   e = 20533068526812417782440649671539166989231638639623145517026 3
   ```

   The intercepted ciphertext is:

   ```
   c = 9937940002828306546302361871389867913555861548411360808100 82
   ```

   Your task: Decrypt the message, and use the integrification process from class to convert the decrypted integer back into the book's title. What book contains the secret?