Lectures 19 and 20

We described RSA, which requires is based on two primes p, and q, and two numbers ϵ and δ such that $\epsilon\delta \equiv 1 \bmod (p-1)(q-1)$.

Then Alice encodes M by $M^\epsilon \bmod pq$, and Bob decodes by M^ϵ by $(M^\epsilon)^\delta \bmod pq$.

We showed $(M^\epsilon)^\delta = M^{\epsilon\delta} \equiv M \bmod pq$.

We also showed that RSA depends on having a fast method to exponentiate in **Z**, and introduced the repeated squaring algorithm.

We also introduced the big-O and little-o notation to easily compare growth rates.

Exercises for Lectures 19 and 20

1. Suppose we have an RSA scheme in which $p = 13$ $q = 17$. Suppose Alice's encoding key is 19. What is the Bob's decoding key?

   How many possible encoding keys could Alice have been assigned. (Hint - use inclusion, exclusion.)

2. Suppose we have an RSA scheme in which $p = 101$ $q = 103$. How many possible encoding keys are there?

3. Suppose we have an RSA scheme in which $p = 41$ $q = 43$. Can we use 41 or 43 as encoding keys? If so, what are the decoding keys?

4. Compute $2^{1000} \bmod 11$.

5. Compute $2^{1000} \bmod 101$.

6. Use fast exponentiation to compute $10^{18} \bmod 13$.

7. Use fast exponentiation to compute $10^{17} \bmod 101$.

8. Show that the number of binary digits of the number n is $O(\log_2(n))$.

9. Show that the number of binary digits of the number n is $O(\log_{10}(n))$.

10. Show $\ln(n) = O(n \ln(n))$.

11. Show $\ln(n) = o(n \ln(n))$.

12. Is $e^n = O(2^n)$?

13. Is $n^3 = O(n^4)$?

14. Let $f(n)$ be a quadratic and $g(n)$ be a cubic.

    Show $f(n) + g(n) = O(n^3)$.

    Show $f(n)g(n) = O(n^5)$.

    Show $g(n)/f(n) = O(n)$.

    Show $f(g(n)) = O(n^6)$.