Cryptography Midterm Problems

Carmichael Numbers (Problem)

Fermat's little theorem gave us a way to think about whether a number is prime or not without factoring it. Compute $2^{N-1} \pmod N$, $3^{N-1} \pmod N$, $5^{N-1} \pmod N$, and $7^{N-1} \pmod N$ for $N = 1729$. What can you conclude? What happens if you try to factor 1729 using a pocket calculator?

Find out about pseudoprimes or Carmichael numbers in a book on elementary Number Theory or by searching the web. Explain what they are and find at least one example of a Carmichael number. What is it about a Carmichael number that makes it "behave like" a prime? What mathematical conditions must be satisfied for a number to be a Carmichael number? Show how these conditions lead to the behavior that mimics true primes.

Divisibility Tests (Problem)

We know from experience that the numbers that are divisible by 2 are exactly those whose last digit is 0, 2, 4, 6 or 8. Similarly, a number is a multiple of 5 if its last digit is a 0 or a 5. Using modular arithmetic we can explain why these divisibility tests work. Just as we did when we talked about check digits we can expand a number N in powers of 10:

$$54237915 = 5 \times 107 + 4 \times 106 + 2 \times 105 + 3 \times 104 + 7 \times 103 + 9 \times 102 + 1 \times 10 + 5$$

Since 10 is equivalent to 0 (mod 5), this expansion shows that that 54237915 = 5 = 0(mod 5). In other words, N is a multiple of 5. There are many other divisibility tests you may have seen before. A number is divisible by 3 if and only if the sum of its digits is a multiple of 3. For example 1267221 is a multiple of 3 because the sum of its digits is 21 and 21 is a multiple of 3. Use modular arithmetic and the expansion of N into powers of 10 to explain why this test works. A similar test tells you if N is divisible by 11. The test for divisibility by 11 is based on the fact that 10 = -1(mod 11). Figure out what this test is and show why it works. There is also a test for divisibility by 7 and 13 based on the fact that $1001 = 7 \times 11 \times 13$. Can you figure out this test and explain how it works? Try to find a test for divisibility by 17. Can you explain how Fermat's little theorem gives a divisibility test for any prime p?

Pseudorandom Digits Generation (Problem)

Very often, scientists need random numbers for their computer programs, for example,

when they are simulating random processes that occur in nature. Since computers don't actually do anything randomly, it is impossible to actually get a random number from a computer. But for many applications, a set of numbers that seem random are good enough, so we settle for that. A computer typically has a pseudorandom number generator, which is just a program that generates numbers that seem random in many respects, but are not, in fact, random.

Let us use modular arithmetic to try to generate a sequence of random digits. We will use the prime number p = 1069 in our computation. We will use the number 342 as our "seed," or starting point. Now we generate the our first three digits. We calculate seed $\times$ seed (modulo p) = 342 $\times$ 342 (mod 1069) = 443. Then we generate the next three digits by multiplying the result 443 by the seed 342 modulo p, and we will get 777. Then we continue: we multiply each result by the seed modulo 1069 to get the next 3-digit entry of the sequence. In case we obtain a number below 100, we add leading zeros to get three digits total. In case we get a number more than 999, we disregard this number and continue (that is, we do not use it in our sequence, but we still multiply it with 342 to obtain the next 3-digit number).

Explain why we want to add leading zeros to numbers with fewer than 3-digits, and explain why we disregard numbers with more than 3 digits. That is, explain why the digits would be less uniformly distributed if we didn't follow these procedures.

Explain why this pseudorandom digit generator will always end up generating digits in cycles. (That is why it is a pseudo-random digit generator.)

Try different seeds. Find seeds that generate cycles of different lengths. Explain how we should try to choose seeds to make the cycle longer. Explain which seeds we need to avoid, so as not to get easy patterns.

Primality Tests (Problem)

Wilson's Theorem (1770) states that if n is an integer greater than one, then n is prime if and only if $(n-1)! = -1 (\text{mod } n)$. We could use this as a test to see whether a number is prime, but $(n-1)!$ is difficult to calculate as compared to a^{n-1}, so primality tests based on Fermat's little theorem are much more practical.

Explain why Wilson's theorem is true.

If n is an integer greater than one and n is not prime, what do we get when we reduce $(n-1)!$ modulo n? Make an estimate of how much time proving primality with Wilson's theorem can take.

On the page `http://www.utm.edu/research/primes/prove/index.html` you can find a discus- sion of primality testing.

This web site contains several primality tests that are good for special cases, that is,

when the candidate- prime number has a special form. Pick one of the tests and explain how it works.

The site also gives some probabilistic tests. Pick one of them and explain how it works.

Search the Internet and find the biggest known prime. How many digits does it have? What is special about this number?

Explain how it is possible for the biggest known prime to have many more digits than the largest numbers that present-day factoring algorithms can handle.

Secret Sharing Schemes (Essay)

Imagine the government trusts you and 15 other people to work on a secret project this Saturday. To enter the extreme-security building you work in, you and your co-workers must enter a secret number, say X, into a computer on the door of the building. You get a secret key (which is different from X), and your co-workers get their own secret keys. Because the government doesn't trust any one of you on your own, or even just two or three of you together, security demands that there must be at least 11 people together, with their 11 different keys, to get access to the building. (Allowance is made for the possibility that up to 4 people cannot make it, because of illness or other reasons.) The idea is that you and those 10 other people must carry out a computation first, together, to figure out the secret key to the door. The computation won't work without all 11 keys. (Note however, that it should work whenever 11 different people among the 15 get together!) What sort of computation should the government set up for you to do?

Here's a quote from the website `http://www.rsasecurity.com`

Secret sharing schemes were discovered independently by Blakley and Shamir. The motivation for secret sharing is secure key management. In some situations, there is usually one secret key that provides access to many important files. If such a key is lost (for example, the person who knows the key becomes unavailable, or the computer which stores the key is destroyed), then all the important files become inaccessible. The basic idea in secret sharing is to divide the secret key into pieces and distribute the pieces to different persons in a group so that certain subsets of the group can get together to recover the key.

The security scheme of Shamir is based on polynomial interpolation. Please describe in detail how and why this scheme works, and give an example of a set of 4 keys which together encodes a message using Shamir's scheme. If any of these keys are missing, one should not be able to decode the message! For some good descriptions of this scheme, please check:

- `http://www.rsasecurity.com/rsalabs/faq/2-1-9.html`
- `http://www.rsasecurity.com/rsalabs/faq/3-6-12.html#`