CS431 Cryptography Fall 2016

Homework 3

Due: Lesson 18

(50 pts)

1. [5] Use the Euclidean Algorithm to compute GCD(2310, 163). Use that fact to help prove 163 is prime. Hint: 2310 = 2\*3\*5\*7\*11. (Interesting fact: if you want to know whether any small primes divide a number, this is often a faster way to check, rather than dividing by lots of small primes).

GCD(2310, 163) = 1 (Excel). This can be used to prove that 163 is prime because 2310 is

a composite number. We can use the prime roots of 2310 to check the primality of 163. Given that 2310 and 163 are relatively prime, the prime factors of 2310 are not factors of 163. We can check the rest of the primes less than 163 by finding the GCD of the product of ‘these primes. In doing this, the GCD is 1. Thus, 163 is prime.

2. [5] Evaluate the following expressions. Use the algorithm in section 3.5 of the text to do this efficiently, and show your work.

a. *x ≡* 3105 (mod 19)

b. *x ≡* 5246903 (mod 27)

1. 32 *≡* 9, 34 *≡* 5, 38 *≡* 6, 316 *≡* 17, 332 *≡* 4, 364 *≡* 16. 105 = 64+32+8+1 🡪

16\*4\*6\*3*≡* 12 (mod 19).

1. 52 *≡* 25, 54 *≡* 4, 58 *≡* 16, 516 *≡* 13, 532 *≡* 7, 564 *≡* 22, 5128 *≡* 25, 5256 *≡* 4, 5512 *≡* 16,

51024 *≡* 13, 52048 *≡* 7, 54096 *≡* 22, 58192 *≡* 25, 516384 *≡* 4, 532768 *≡* 16, 565536 *≡* 13, 5131072 *≡* 7.

246903 = 131072+65536+32768+16384+1024+64+32+16+4+2+1 🡪 7\*13\*16\*4\*13\*22\*7\*13\*4\*25\*5 *≡* 8 (mod 27).

3. [5] Find the last three digits of 31201576

To do this, I will conduct the process above pertaining to x *≡* 31201576 (mod 1000).

After running a Python program to calculate this, I found x = 921.

4. [5] Suppose p and q are distinct primes, and gcd(m, pq)>1. Prove that m is a multiple of p or a multiple of q, or both. (Hint: What are the factors of pq?)

If gcd(m, pq) > 1, then m is a multiple of p or a multiple of q, or both. The factors of pq are p and q. If the gcd of(m, p) > 1, then m is a multiple of p as there any other number would have a gcd = 1. Similarly, if the gcd(m, q) > 1, then m is a multiple of q. When p and q are multiplied together, if the gcd(m, pq)>1, then m is a multiple of p, q, or both due to the fact that p and q are distinct primes. If m was not a multiple of p, q, or both, then the gcd(m, pq) would equal 1 through the general properties of the greatest common denominator.

5. [5] The ciphertext 7591 came from RSA with modulus n=8549=103\*83 and e=4771. Find the ASCII symbol of the plaintext. Show all work.

Φ(8549)=82\*102=8364. d=e-1(mod 8364) = 4111. M *≡* Cd *≡* 75914111 (mod 8549) *≡* 236 🡪 ì

6. [5] Bob decides to make RSA stronger by double encrypting his messages. So he first encrypts his message with key e1, and then encrypts that ciphertext with e2. Alice decrypts with d2 then d1. Is this more secure than single encryption? Why or why not? Assume all keys are the same size (i.e. same number of bits) and that n is unchanged.

This is not more secure than single encryption. By doing this, the p and the q of the final ciphertext message are not primes. Thus, because of this the key can be decrypted a single time through the modular inverse of the totient of the final n. Essentially, because the n does not change, the final ciphertext is no more secure given a double encryption than it is from single encryption.

7. [5] A toy public key cryptosystem called “Kid Krypto” works in this fashion: To be able to receive enciphered messages from others elsewhere, Ursula chooses four positive integers *a*, *b*, *A*, and *B* and calculates

*M = ab –* 1

*e = AM + a*

*d = BM + b*

*n = (ed – 1)/M*

She then publicizes *e* and *n* with instructions that a numerical message *x* in the range 0 to *n*-1 is to be enciphered as

*y = e\*x* mod *n*

Ursula deciphers a message *y* by computing

*x = d\*y* mod *n*

(a) If Ursula chooses *a*=55, *b*=21, *A*=12, and *B*=6, calculate her values of *M*, *e*, *d*, and *n*.

M = 1154, e = 13903, d = 6945, n = 83671

(b) Suppose that Ursula had used other choices of *a*, *b*, *A*, and *B* to generate the values of *n*=25123 and *e*=2273. Further, suppose that Eve intercepts an enciphered PIN 12255 belonging to another bank customer, Walter, and would like to decipher it to gain access to his account. If all Eve knows is the relationship among *e*, *d*, and *n*, what can she do to find *d*? What alternative approach might she take to find an encrypted PIN without applying the standard decrypt formula? Find Walter’s PIN.

Eve can use the fact that d and e are inversely related pertaining to x and y mod n. Thus, Eve can use the known values of e, n, and y to find d. As an alternative to the standard decrypt, she can use the encryption formula and multiple y by the multiplicative inverse of e, thus giving the equation y\*e-1 = x mod n. Using this method, 12255\*2273 = 2017 mod n.

*8.* [5] Use the quadratic formula to determine the existence of roots of the polynomial 2x2 + 18x + 28 (mod 739). If there are roots, give them and show that your answers are correct (i.e.: evaluate the expression and show it is equivalent to 0 mod 739). If there are no roots, explain why. Remember that to answer this question, the mathematical operations in the quadratic formula must be performed modulo 739. How do you know that you have found all the roots?

The roots are x = 732 + k\*739 and x = 737 + k\*739. 2\*(732^2)+18\*732+28 = 1084852 % 739 = 0. Also, 2\*(737^2)+18\*737+28 = 1099632 % 737 = 0. I know I have found all the roots as given that the program I wrote found no other possible roots below 739, the equations for the roots above give all the continuous roots for the quadratic equation.

9. [10] Write a small program that uses Fermat’s Little Theorem to print out the numbers from 1 to *n* (input by the user) that are “probably” prime. Check each number using brute force and indicate when Fermat’s is wrong. Output should look similar to the following:

Find primes up to what number? **500**

…

328

329

330

331 Prime & Fermat Pseudoprime

332

333

334

335

336

337 Prime & Fermat Pseudoprime

338

339

340

341 \*\*\*\*\* NOT PRIME but Fermat Pseudoprime \*\*\*\*\*

342

343

344

345

346

347 Prime & Fermat Pseudoprime

348

349 Prime & Fermat Pseudoprime

350

351

…

Email your code (source and executable) and a screen shot showing the lowest composite Fermat Pseudoprime you found. A number *p* is a Fermat Pseudoprime if some *a* such that 1< *a* < *p* with (a,p) coprime exists such that 𝑎𝑝−1≡1 (𝑚𝑜𝑑 𝑝).

