Fall 2019 CIS 3362 Homework #5 Solutions
Number Theory, RSA

1) What is the prime factorization of 1337834957760?

Solution
The prime factorization can be found by dividing by the smallest prime number until it is no longer a factor, and then continuing with each consecutive prime:

2 x 2 x 2 x 2 x 2 x 2 x 3 x 3 x 3 x 3 x 5 x 7 x 7 x 1053347

= $\mathbf{2^6 \times 3^4 \times 5 \times 7^2 \times 1053347}$

2) What is φ(1337834957760)?

Solution
Using the above prime factorization:

$\varphi(2^6) \times \varphi(3^4) \times \varphi(5) \times \varphi(7^2) \times \varphi(1053347)$

$(2^6-2^5)\text{x}(3^4-3^3)\text{x}(5-5^0)\text{x}(7^2-7)\text{x}(1053347-1053347^0)$

(32)x(54)x(4)x(42)x(1053347) = **305790557184**

3) Using Fermat's Theorem, determine the remainder when 135^{2672} is divided by 179.

Solution
Because 179 is prime, 135 and 179 are relatively prime.
By the definition of relative primality:
$135^{178} \equiv 1 \pmod{179}$
Using Fermat's theorem:
$135^{2672} \equiv (135^{178})^{15} \times 135^2 \equiv 1 \times 18225 \equiv$ **146** (mod 179)

4) Using Euler's Theorem, determine 7429^{628993} mod 529984.

Solution
7429 is only divisible by 17 and 437, neither of which is a divisor of 529984, thus they are relatively prime.
By Euler's Theorem we can state
$7429^{\varphi(529984)} \equiv 1 \pmod{529984}$
First the prime factorization of 529984:
$2^6 \times 7^2 \times 13^2$
$\varphi(2^6) \times \varphi(7^2) \times \varphi(13^2) = (2^6 - 2^5)(7^2 - 7^1)(13^2 - 13^1) = 209664$
$7429^{628993} \equiv 7429^{628992 + 1} \equiv (7429^{209664})^3 + 7429^1 \equiv 1^3 + 7429 \equiv$ **7429** (mod 529984)

5) In an RSA scheme, p = 31, q = 19 and e = 77. What is d?

Solution
First we must find $\varphi(31 \times 19) = \varphi(31) \times \varphi(19) = 30 \times 18 = 540$

$d = 77^{-1} \pmod{540}$

Using the Extended Euclidean Algorithm:

$540 = 77 \times 7 + 1$ (shortest Euclidean ever)

$540 \times 1 – 77 \times 7 = 1 \pmod{540}$
$-7 \times 77 = 1 \pmod{540}$
$-7 + 540 = 1 \pmod{540}$

$533 = 1 \pmod{540}$

d = **533**

6) A primitive root, α, of a prime, p, is a value such that when you calculate the remainders of α, $\alpha^2, \alpha^3, \alpha^4, \ldots, \alpha^{p-1}$, when divided by p, each number from the set {1, 2, 3, ..., p-1} shows up exactly once. Prove that a prime p has exactly φ(p-1) primitive roots. In writing your proof, you may assume that at least one primitive root of p exists. (Normally, this is the first part of the proof.) (Note: This question is difficult, so don't feel bad if you can't figure it out.)

Solution
We assume that at least one primitive root exists. Let's call this α. We know that of the p-1 values 1, 2, 3, …, p-1, exactly φ(p-1) of them share no common factor with p-1, based on the definition of φ.

In order to prove the assertion, we must prove that α^k is a primitive root if and only if gcd(k, p-1) = 1. If we can prove this, then from the list $\alpha, \alpha^2, \alpha^3, \alpha^4, \ldots, \alpha^{p-1}$, the terms that are primitive roots are precisely the terms with the exponents that don't share a common factor with p-1, of which there are exactly φ(p-1).

Let gcd(k, p – 1) = 1. We will prove that α^k is a primitive root. We prove this using proof by contradiction. Assume the opposite, that α^k is NOT a primitive root. Then, we must have that two values in the list $\alpha^k, \alpha^{2k}, \alpha^{3k}, \ldots, \alpha^{k(p-1)}$ that are equivalent mod p. Let these two values be α^{ik} and α^{jk}, where $0 < i < j < p$. Thus, we have:

$\alpha^{jk} \equiv \alpha^{ik} \mod p$
$\alpha^{jk} - \alpha^{ik} \equiv 0 \mod p$
$\alpha^{ik}(\alpha^{jk-ik} – 1) \equiv 0 \mod p$

We know that p shares no common factors with α^{ik}.

It follows that $p \mid \alpha^{jk-ik} - 1$. Thus

$\alpha^{jk-ik} - 1 \equiv 0 \mod p$
$\alpha^{(j-i)k} \equiv 1 \mod p$

Since α is a primitive root, we know that the exponent on the left must be a multiple of $p - 1$:

$(p - 1) \mid (j - i)k$.

We know that $\gcd(p - 1, k) = 1$. Thus it follows that $(p - 1) \mid (j - i)$. But this contradicts the fact that $0 < i < j < p$, which means that $i >= 1$, $j <= p-1$, so $j - i > 0$ and $j - i \leq p - 2$.

This is our contradiction. It follows that our initial assumption that two values on the given list were equivalent mod p is faulty. If no two of these values are equivalent mod p, we can conclude that α^k is a primitive root.

Now, the second part of the proof is that if $\gcd(p - 1, k) > 1$, then α^k is NOT a primitive root. Let $c = \gcd(p - 1, k) > 1$. Now, consider the term $(\alpha^k)^{(p-1)/c}$ mod p. The exponent $(p-1)/c$ is clearly less than $p - 1$. Secondly, this is equivalent to $\alpha^{k/c * (p-1)}$ mod p. Notice that c divides evenly into k because $c = \gcd(k, p - 1)$, thus c is a divisor of k. Let $m = k / c$, and $m \in Z$. Thus $\alpha^{k/c * (p-1)} \equiv (\alpha^{p-1})^{/c} \equiv 1^m \equiv 1 \ (mod\ p)$. This means that α^k isn't a primitive root since raising it to a power less than $p - 1$ yields 1.

Thus, we have shown that if AND only if $\gcd(p - 1, k) = 1$, is α^k a primitive root of p. Thus, to count the number of primitive roots, we simply look at the list $\alpha, \alpha^2, \alpha^3, \alpha^4, \ldots, \alpha^{p-1}$ and count the number of terms that have exponents relatively prime to $p - 1$. By definition of α, this number is exactly $\varphi(p - 1)$. As a concrete example, if we know that 2 is a primitive root of $p = 19$, it follows that $2^1, 2^5, 2^7, 2^{11}, 2^{13}$, and 2^{17} are all primitive roots of 19, since 1, 5, 7, 11, 13 and 17 don't share any common factors with 18, $p - 1$.

Note: This solution is written by Sushant Kulkarni, a past TA of the course.

7) One of the primitive roots (also called generators) mod 43 is 29. There are 11 other primitive roots mod 43. One way to list these is 29^{a1} mod 41, 29^{a2} mod 41, … 29^{a12} mod 41, where $0 < a1 < a2 < … < a12$. (Note: it's fairly easy to see that a1 = 1, since 29 is a primitive root.) Find the values of a10, a11 and a12 and the corresponding remainders when 29^{a10}, 29^{a11} and 29^{a12} are divided by 43.

Solution

The key idea here is from the proof of question 6, which shows that if g is a generator mod p, g^x is also a generator if and only if gcd(x, p-1) = 1. Thus, the corresponding exponents, a1, a2, a3, …, refer to the 12 values that are relatively prime to 42. These 12 values are: 1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, and 41. Thus,

a10 = 31, a11 = 37, and a12 = 41

Now, use these three values and fast modular exponentiation to calculate the following three generators mod 43:

29^{31} (mod 43) ≡ 30 (mod 43)
29^{37} (mod 43) ≡ 28 (mod 43)
29^{41} (mod 43) ≡ 3 (mod 43)

8) In the Diffie-Hellman Key Exchange, let the public keys be p = 43, g = 20, and the secret keys be a = 25 and b = 29, where a is Alice's secret key and b is Bob's secret key. What value does Alice send Bob? What value does Bob send Alice? What is the secret key they share?

Solution

a) Alice sends Bob: g^a (mod p)

20^{25} (mod 43) ≡ 3 (mod 43)

b) Bob sends Alice: g^b (mod p)

20^{29} (mod 43) ≡ 34 (mod 43)

c) Shared key: (answer a)$^{\mathbf{b}}$ (mod p) or (answer b)^a (mod p)

3^{29} (mod 43) ≡ 18 (mod 43)

34^{25} (mod 43) ≡ 18 (mod 43)

9) For this question, you are going to implement a RSA protocol to send the TAs and me (Arup) a message. For our RSA system, the public keys are as follows:

n = 1359662499348132121870942313 81
e = 437623485647823657465674567

Your message must be in Radix-64. Please google this format. It allows for 64 characters, encoding each with 6 bits. The characters are: all lowercase letters, all uppercase letters, all digits, the plus sign(+) and a forward slash (/).

First, type your message in a textfile only using those 64 characters. Type 16 characters per line. To encrypt, you will encrypt each line, one by one. Please pad the last line with '+' characters as needed. Convert each line of 16 Raxix-64 characters to a 96 bit integer. This will be your plaintext for RSA. Use the public keys given above and calculate the ciphertext, which will be a number from 1 to n-1. Output this number to a textfile. Do this for each line of the message. Here is what you need to turn in for this question:

1. Your code.

2. A text file with your ciphertext. This should have one number per line, for each block of 16 Radix-64 characters.

If you did everything to specification, the TAs and I should be able to read your message. Please keep it clean =) You may address any one of the three of us in your message, or all three of us, if you'd like!

Solution
A sample code file, plaintext message and encrypted message are posted with this solution in the following files:

sendmsg.py
msg.txt
msg.out

Also, the grading "script", readmsg.py, is included.