CS 70 Discrete Mathematics and Probability Theory
Fall 2026 Hug, Sabin

DIS 3

Modular Arithmetic Intro II

Note 6

Euclid's GCD Algorithm: An algorithm to find $\gcd(x,y)$ efficiently, using the following two identities:

- $\gcd(x,y) = \gcd(y,x)$
- $\gcd(x,y) = \gcd(y, x \bmod y)$

Extended Euclid's GCD Algorithm: An extension to Euclid's GCD algorithm allowing us to find coefficients a and b such that $ax+by = \gcd(x,y)$. Note that if $\gcd(x,y) = 1$, then the equation $ax+by = \gcd(x,y) = 1$ tells us that (a,x) are inverses in $\pmod y$ and (b,y) are inverses in $\pmod x$.

Chinese Remainder Theorem: Given a system of k modular equations $x \equiv a_i \pmod{n_i}$, for various constants a_i and coprime moduli n_i, there exists a *unique* solution x defined as follows:

$$x \equiv \sum_{i=1}^{k} a_i b_i \pmod N$$

$$b_i = \left(\frac{N}{n_i}\right)\left(\left(\frac{N}{n_i}\right)^{-1} \bmod n_i\right)$$

$$N = \prod_{i=1}^{k} n_i$$

1 Music Video

Note 6

Xander, a music artist, is wrapping up his latest single, "Remainders Everywhere."

(a) Xander decides to run a choreography contest for his new song. The three finalists (Alice, Bob, and Candy) have very different rhythms for their choreographies though! Alice created her dance in groups of 3 beats with a 2 beat extra finale at the end. Bob's submission is 5 beat groups with a 1 beat finale. Candy's is 11 beat groups and has a 7 beat finale. If all submissions perfectly match the song, what is the smallest possible number of beats in Xander's original song?

(b) Now that Xander's music video and choreography is ready, he wants to send the final result to his mentor Yonathan to take a look. But, he needs to avoid letting the song fall into the hands of his fans, who are eagerly waiting a leak. So, Xander decides to use the RSA

cryptosystem to send it to Yonathan. Using primes $p = 11$ and $q = 17$, he uses a public key of $(N, e) = (187, 7)$. What should their private key be?

Solution:

(a) If we let x be the number of beats in Xander's song, then the problem tells us that

$$x \equiv 2 \pmod 3, \quad x \equiv 1 \pmod 5, \quad x \equiv 7 \pmod{11}.$$

Since 3, 5, and 11 are pairwise coprime, we can use the Chinese Remainder Theorem to get an answer for x in modulo $N = 3 \cdot 5 \cdot 11 = 165$.

First to find the multiplicative inverses, we can find that the inverse of 15 modulo 11 is 3, the inverse of 33 modulo 5 is 2, and the inverse of 55 modulo 3 is 1. So, the answer is $x = 2 \cdot 55 \cdot 1 + 1 \cdot 33 \cdot 2 + 7 \cdot 15 \cdot 3 = 491$ modulo 165. Thus, the smallest x is 161 beats.

(b) First, we see that $(p-1)(q-1) = 10 \cdot 16 = 160$. The private key d will be the inverse of $e = 7$ in modulo 160. Since $7 \cdot 23 = 161 \equiv 1 \pmod{160}$, we see that $d = 23$.

2 Sparsity of Primes

Note 6

A prime power is a number that can be written as p^i for some prime p and some positive integer i. So, $9 = 3^2$ is a prime power, and so is $8 = 2^3$. $42 = 2 \cdot 3 \cdot 7$ is not a prime power.

Prove that for any positive integer k, there exists k consecutive positive integers such that none of them are prime powers.

Hint: This is a Chinese Remainder Theorem problem. We want to find n such that $(n+1)$, $(n+2)$, ..., and $(n+k)$ are all not powers of primes. We can enforce this by saying that $n+1$ through $n+k$ each must have two distinct prime divisors. In your proof, you can choose these prime divisors arbitrarily.

Solution:

We want to find n such that $n+1, n+2, n+3, \ldots, n+k$ are all not powers of primes. We can enforce this by saying that $n+1$ through $n+k$ each must have two distinct prime divisors. So, select $2k$ primes, $p_1, p_2, \ldots, p_{2k}$, and enforce the constraints

$$\begin{aligned}
n+1 &\equiv 0 \pmod{p_1 p_2} \\
n+2 &\equiv 0 \pmod{p_3 p_4} \\
&\vdots \\
n+i &\equiv 0 \pmod{p_{2i-1} p_{2i}} \\
&\vdots \\
n+k &\equiv 0 \pmod{p_{2k-1} p_{2k}}.
\end{aligned}$$

By Chinese Remainder Theorem, we can calculate the value of n, so this n must exist, and thus, $n+1$ through $n+k$ are not prime powers.

What's even more interesting here is that we could select any $2k$ primes we want!

3 Debugging the Chinese Remainder Theorem

Note 6

Suppose we want to solve the system

$$x \equiv 2 \pmod 5, \quad x \equiv 3 \pmod 7, \quad x \equiv 4 \pmod 9.$$

Since 5, 7, and 9 are pairwise coprime, we can use the Chinese Remainder Theorem with

$$N = 5 \cdot 7 \cdot 9 = 315.$$

Recall that in the CRT construction, each b_i should satisfy

$$b_i \equiv 1 \pmod{n_i}$$

and

$$b_i \equiv 0 \pmod{n_j}$$

for every $j \neq i$.

A student computes

$$b_1 = 126, \quad b_2 = 90, \quad b_3 = 280.$$

(a) Check each of b_1, b_2, and b_3 against the required CRT properties. Which value, if any, is incorrect? Explain why the incorrect value fails.

(b) Compute a correct replacement for the incorrect value.

(c) Using the corrected values of b_1, b_2, and b_3, find the solution x modulo 315. Verify that your answer satisfies all three original congruences.

Solution:

(a) For $b_1 = 126$,

$$126 \equiv 1 \pmod 5,$$

and

$$126 \equiv 0 \pmod 7, \quad 126 \equiv 0 \pmod 9.$$

Thus, b_1 is correct.

For $b_2 = 90$,

$$90 \equiv 0 \pmod 5, \quad 90 \equiv 6 \pmod 7, \quad 90 \equiv 0 \pmod 9.$$

Since b_2 should satisfy

$$b_2 \equiv 1 \pmod 7,$$

$b_2 = 90$ is incorrect.

For $b_3 = 280$,

$$280 \equiv 0 \pmod 5, \quad 280 \equiv 0 \pmod 7, \quad 280 \equiv 1 \pmod 9.$$

Thus, b_3 is correct.

Therefore, the incorrect value is

$$\boxed{b_2 = 90}.$$

(b) We want

$$b_2 = \frac{315}{7}\left(\frac{315}{7}\right)_7^{-1}.$$

Thus,

$$b_2 = 45(45_7^{-1}).$$

Since

$$45 \equiv 3 \pmod 7$$

and

$$3^{-1} \equiv 5 \pmod 7,$$

we get

$$b_2 = 45 \cdot 5 = \boxed{225}.$$

(c) Using the CRT construction,

$$x \equiv 2b_1 + 3b_2 + 4b_3 \pmod{315}.$$

Substituting the corrected values,

$$x \equiv 2(126) + 3(225) + 4(280) \pmod{315}.$$

Therefore,

$$x \equiv 252 + 675 + 1120 \pmod{315},$$

so

$$x \equiv 2047 \pmod{315}.$$

Reducing modulo 315,

$$\boxed{x \equiv 157 \pmod{315}}.$$

We verify:

$$157 \equiv 2 \pmod 5,$$
$$157 \equiv 3 \pmod 7,$$

and

$$157 \equiv 4 \pmod 9.$$

Thus, the solution satisfies all three original congruences.

4 Euclid's Gallons

Note 6

You have two jugs that hold a and b gallons, a tap with unlimited water, and no markings on the jugs. The only moves allowed are: fill a jug to the top, empty a jug completely, or pour one jug into the other until the first is empty or the second is full. The goal is to end with exactly t gallons in one of the jugs.

(a) (Famous one!) With a 3 gallon and a 5 gallon jug, get exactly 4 gallons. Write out your moves. Then count how many times you filled each jug and emptied each jug, and try to write an equation that must hold for your solution.

(b) Call pouring from one jug into another "pumping a into b." In one round of pumping, we fill the a jug and pour it into the b jug. If the b jug fills up before the a jug is empty, empty the b jug and pour the leftover from the a jug into it. Every round starts and ends with the a jug empty. Try pumping 3 into 5 for a few rounds and keep track of how much the 5 jug holds after each round.

Now suppose $\gcd(a,b) = 1$ and $0 < t < b$. Show that the b jug holds exactly t gallons after some round, and give a formula for which round this happens on.

(c) You have a 7 gallon and a 12 gallon jug and need 5 gallons. Use the extended Euclidean algorithm to find how many rounds it takes to pump 7 into 12. Then find how many rounds it takes to pump 12 into 7 (same procedure with the roles swapped: fill the 12 jug, pour into the 7 jug, empty the 7 jug when it fills). Which would you rather do?

(d) With a 6 gallon and a 9 gallon jug, can you measure 4 gallons? Explain.

(a) **Solution:** Fill 5, pour into 3 (the 5 jug has 2 left), empty 3, pour the 2 into 3, fill 5, pour into 3 until it is full (the 5 jug has $5-1=4$). This uses 2 fills of the 5 jug and 2 empties of the 3 jug, and $2\cdot 5 - 2\cdot 3 = 4$.

Another solution: fill 3, pour into 5, fill 3, pour into 5 (the 3 jug has 1 left), empty 5, pour the 1 into 5, fill 3, pour into 5. This uses 3 fills of the 3 jug and 1 empty of the 5 jug, and $3\cdot 3 - 1\cdot 5 = 4$.

In general, the equation

$$(\text{fills of } 5)\cdot 5 + (\text{fills of } 3)\cdot 3 - (\text{empties of } 5)\cdot 5 - (\text{empties of } 3)\cdot 3 = 4$$

is true for any solution. Track the total amount of water in the two jugs combined. Filling a jug adds its capacity to the total, emptying a jug subtracts its capacity, and pouring just moves water between the jugs without changing the total. The total starts at 0, so at the end it equals the left hand side above. At the end one jug holds 4 and the other holds 0, so the total is 4.

(b) **Solution:** Pumping 3 into 5:

after round	1	2	3	4	5
5 jug holds	3	1	4	2	0

That is $3k \bmod 5$ after round k. In general, after k rounds the b jug holds $ka \bmod b$. We can see this by noting that after k rounds we have filled the a jug k times, so ka gallons went in, and we have emptied the b jug some number of times, throwing away b gallons each time. So the b jug holds $ka - (\text{something}) \cdot b$. Also, the b jug holds between 0 and $b-1$ gallons: the a jug is empty, and the b jug cannot be full, since the round would have emptied it. The only number of the form $ka - (\text{something}) \cdot b$ in that range is $ka \bmod b$.

So the b jug holds t after round k exactly when

$$ka \equiv t \pmod{b}.$$

Since $\gcd(a,b) = 1$, a has an inverse mod b. Multiplying both sides by a^{-1} gives

$$k \equiv t \cdot a^{-1} \pmod{b}.$$

Take $k = (t \cdot a^{-1}) \bmod b$. This k is between 0 and $b-1$, and it is not 0: if it were, then $t \equiv 0 \cdot a = 0 \pmod{b}$, but $0 < t < b$. So after round k, where $1 \le k \le b-1$, the b jug holds exactly t gallons.

We can do a sanity check: assume from (a) $a = 3$, $b = 5$, $t = 4$. We have $3^{-1} \equiv 2 \pmod{5}$ since $3 \cdot 2 = 6 \equiv 1$, so $k = 4 \cdot 2 \bmod 5 = 3$. The table shows the 5 jug holds 4 after round 3. (This is the second solution from part (a): three fills of the 3 jug.)

(c) **Solution:** Run the extended Euclidean algorithm on $(12, 7)$:

$$12 = 1 \cdot 7 + 5, \qquad 7 = 1 \cdot 5 + 2, \qquad 5 = 2 \cdot 2 + 1.$$

Back-substituting,

$$1 = 5 - 2 \cdot 2 = 5 - 2(7-5) = 3 \cdot 5 - 2 \cdot 7 = 3(12-7) - 2 \cdot 7 = 3 \cdot 12 - 5 \cdot 7.$$

From this one equation we can read off both inverses: $7^{-1} \equiv -5 \equiv 7 \pmod{12}$, and $12^{-1} \equiv 3 \pmod{7}$.

Pumping 7 into 12: by part (b), the number of rounds is $k = 5 \cdot 7^{-1} \bmod 12 = 5 \cdot 7 \bmod 12 = 35 \bmod 12 = 11$.

Pumping 12 into 7: the same argument with the roles swapped says the 12 jug holds $12k \bmod 7$ after k rounds, so $k = 5 \cdot 12^{-1} \bmod 7 = 5 \cdot 3 \bmod 7 = 15 \bmod 7 = 1$. Fill the 12 jug, pour into the 7 jug, and the 12 jug has 5 gallons left.

So one run of the extended Euclidean algorithm told us the number of rounds in both directions before we touched a jug, and that one direction is 11 times faster than the other, so we should pour the 12 jug into the 7!

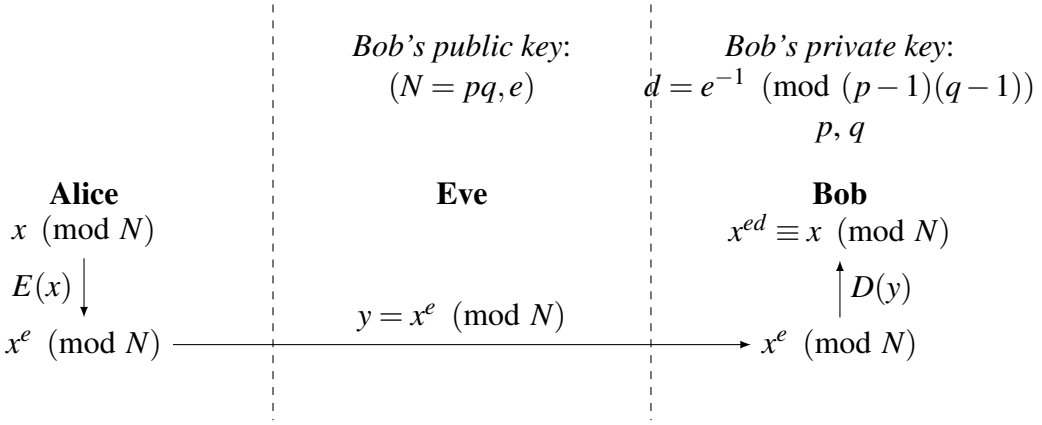
(d) **Solution:** No. Both jugs start with 0 gallons, a multiple of 3. Every move keeps this true: filling puts 6 or 9 in a jug, emptying puts 0, and pouring leaves the jugs holding either $(0, p+q)$ or $(p+q-c, c)$ where p, q are the old amounts and c is a capacity. If p and q were multiples of 3, so are all of these. So each jug always holds a multiple of $\gcd(6,9) = 3$, and 4 is not one.

RSA Intro

Note 7

Fermat's Little Theorem: For all primes p, $a^{p-1} \equiv 1 \pmod{p}$ if $a \neq 0$. An equivalent version of the statement (still assuming p is prime) is $a^p \equiv a \pmod{p}$ for all a.

RSA Scheme: A cryptographic scheme that allows communication over insecure channels via public-key encryption. Alice encrypts her message x with Bob's public key, ensuring that only Bob (with his private key) can decrypt it, which prevents Eve from eavesdropping.



5 RSA for Concert Tickets

Note 7

Alice wants to tell Bob her concert ticket number, m, which is an integer between 0 and 100 inclusive. She wants to tell Bob over an insecure channel that Eve can listen in on, but Alice does not want Eve to know her ticket number.

(a) Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's ticket number is. How did she do it?

(b) Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m? (You may assume that r is coprime to N.)

Solution:

(a) There are only 101 possible values for Alice's ticket number, so Eve can try encrypting all 101 values with Bob's public key and find out which one matches the one Alice sent.

(b) Alice sends $x = r^e \pmod{pq}$, as well as $y = (rm)^e = r^e m^e = xm^e \pmod{pq}$. We can find $x^{-1} \pmod{N}$ using the Extended Euclidean Algorithm, and multiplying this value by y gives us $m^e \pmod{N}$. Now we proceed as in the previous part to find m.

Another approach is to compute xm^e for all 101 values of m, and compare the value to y, checking which one matches.

6 RSA with Multiple Keys

Note 7

Members of a secret society know a secret word. They transmit this secret word x between each other many times, each time encrypting it with the RSA method. Eve, who is listening to all of their communications, notices that in all of the public keys they use, the exponent e is the same. Therefore the public keys used look like $(N_1, e), \ldots, (N_k, e)$ where no two N_i's are the same. Assume that the message is x such that $0 \leq x < N_i$ for every i.

Further, in all of the subparts, you may assume that Eve knows the details of the modified RSA schemes (i.e. Eve knows the format of the N_i's, but not the specific values used to compute the N_i's).

(a) Suppose Eve sees the public keys $(p_1q_1, 7)$ and $(p_1q_2, 7)$ as well as the corresponding transmissions. Can Eve use this knowledge to break the encryption? If so, how? Assume that Eve cannot compute prime factors efficiently. Think of p_1, q_1, q_2 as massive 1024-bit numbers. Assume p_1, q_1, q_2 are all distinct and are valid primes for RSA to be carried out.

(b) The secret society has wised up to Eve and changed their choices of N, in addition to changing their word x. Now, Eve sees keys $(p_1q_1, 3)$, $(p_2q_2, 3)$, and $(p_3q_3, 3)$ along with their transmissions. Argue why Eve cannot break the encryption in the same way as above. Assume $p_1, p_2, p_3, q_1, q_2, q_3$ are all distinct and are valid primes for RSA to be carried out.

(c) Let's say the secret x was not changed ($e = 3$), so they used the same public keys as before, but did not transmit different messages. How can Eve figure out x?

Solution:

(a) Normally, the difficulty of cracking RSA hinges upon the believed difficulty of factoring large numbers. If Eve were given just p_1q_1, she would (probably) not be able to figure out the factors.

However, Eve has access to two public keys, so yes, she will be able to figure it out. Note that $\gcd(p_1q_1, p_1q_2) = p_1$. Taking GCDs is actually an efficient operation thanks to the Euclidean Algorithm. Therefore, she can figure out the value of p_1, and from there figure out the value of q_1 and q_2 since she has p_1q_1 and p_1q_2.

(b) Since none of the N's have common factors, she cannot find a GCD to divide out of any of the Ns. Hence the approach above does not work.

(c) Eve observes $x^3 \pmod{N_1}, x^3 \pmod{N_2}, x^3 \pmod{N_3}$. Since all N_1, N_2, N_3 are pairwise relatively prime, Eve can use the Chinese Remainder Theorem to figure out $x^3 \pmod{N_1N_2N_3}$. However, once she gets that, she knows x, since $x < N_1$, $x < N_2$, and $x < N_3$, which implies $x^3 < N_1N_2N_3$, so she can directly take the cube root of the result from CRT. Uh oh!