Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture - 17
Extended Euclid Algorithm

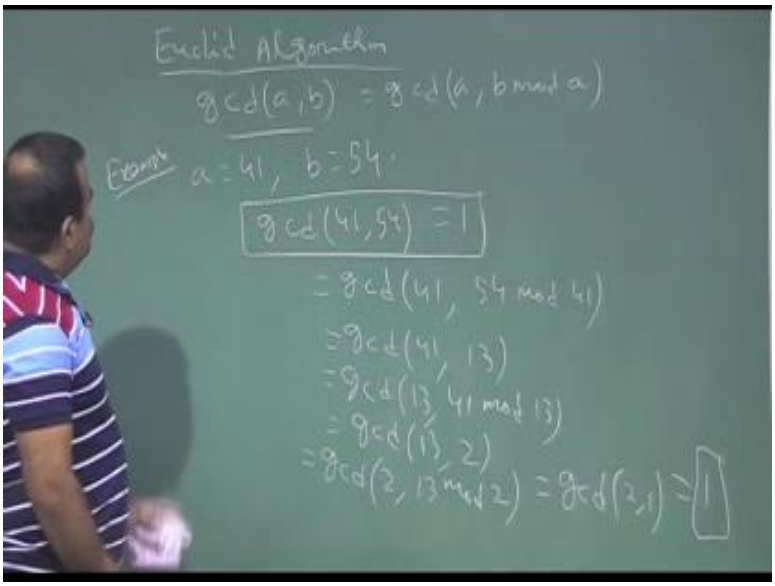
We have seen the multiplicative inverse modulo n or modulo some integer m. So now we will talk about how to find that inverse.

(Refer Slide Time: 00:36)

So, suppose with like n be a prime, I know n be an integer then we have seen the an a is also an integer, be an integer. Then we have seen if a is relatively prime with n; that means, gcd of a and n is 1 then we have seen a as a inverse, this inverse is in the multiplicative sense multiplicative inverse mod n. So that means, there exist b such that a into b is from you into 1 mode n. Then this b is called a inverse mod n. So, this is what is the multiplicative inverse of a under multiplicative inverse of a modulo n.

So, in this lecture we will see how we can find such b, how we can find the inverse a inverse. So, that is basically coming from the Euclid algorithm, but we have to extend it a little bit. Euclid algorithm will give us what ? Gcd of two numbers a b. So, this is basically how we define the inverse of a, and for a distance of inverse we know that this is basically this a must be relatively prime with n.

(Refer Slide Time: 02:51)

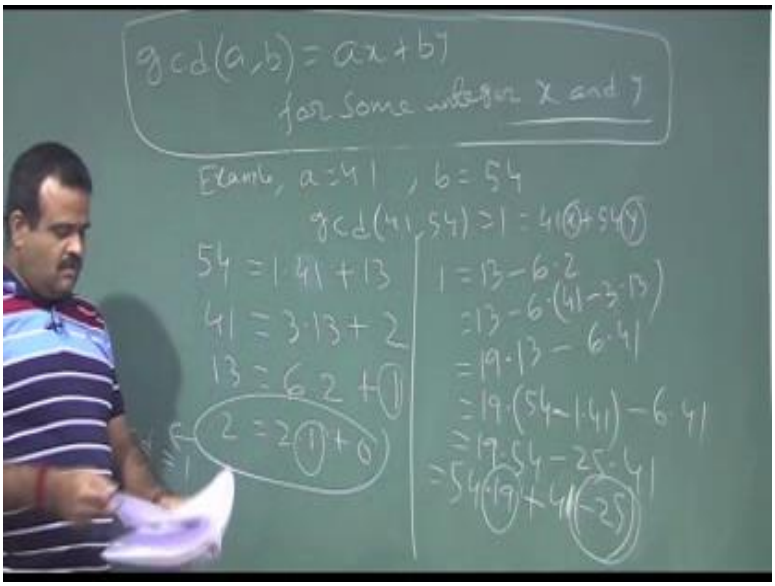


So, we need to know how we can find the inverse. So, this is basically how to find the inverse a inverse. So, this is basically coming from the Euclid algorithm, we know the Euclid algorithm. So, Euclid algorithm is basically finding the gcd of two elements a b; two integer a b. So, how we can find the gcd of two integer a b? Suppose a is a 41, b is a 54.

Now, basically Euclid algorithm is basically gcd of a b same as gcd of a comma b mod a; if b is greater than a. So, this is the theorem we have proved and this theorem is basically the Euclid algorithm; this theorem will give us. So, these are the remainder. So, then we take this remainder with a like this, so this will continue.

So, if you take this example then basically if we try to find out gcd of 41, 54 then this is basically gcd of 41 54 mod 41. Now 54 mod 41 is how much? 54 mod 41 is basically 13. So, then again we take this like this gcd of 13 then 41 mod 13, this is because we can this theorem; gcd of a b is basically the gcd of b a. Anyway so this is basically; 41 mod 13 is basically 2, so gcd of 13 comma 2. Now again we will write like this gcd of 2 comma 13 mod 2. So, this will give us basically 1. So, gcd of 2 comma 1; so this is basically 1. So, this gcd of this is basically 1.

(Refer Slide Time: 05:44)



Now, we will prove a theorem where gcd of a b can be written as; gcd of a b should be written as ax plus by for some integer x and y. So, this is a theorem we need to prove this and this is basically will give us the a this x give us the extremely Euclid algorithm. So now, for this example, how to get this x y? So basically, to get x y, so gcd of this is 1, so 1 must be written as - in our example if we take a is equal to 41 and b is equal to 54, then just now we have seen the gcd of 41 and 54 is 1. That means, we must get to x y such that this should be written as 41 x plus 54 y. So, x y are two integer.

So, how to get such x y that is the question. So, to get of x y we have to just follow the Euclid algorithm that remainder then we will go back like this, so let us try that. First we are finding the 54 mod 41 that means that is 13; so 54 will be written as 1 into 14, I sorry 1 into 41 plus 13. That means gcd of these two is basically same as gcd of 41 and 13. Then again 41 should be written as; then the 41 mod 13 is basically 2. So, 41 should be written as this plus 2.

Again the 13 will be written, now we need to find out that this is the Euclid algorithm. So now we need to find the gcd of this 13 and 2. So, this is basically 6 into 2 plus 1. Now we need to find the gcd of 2 and 1, so this will give us 2 into 1 plus 0. That means, this last step will tell us the gcd basically 1.

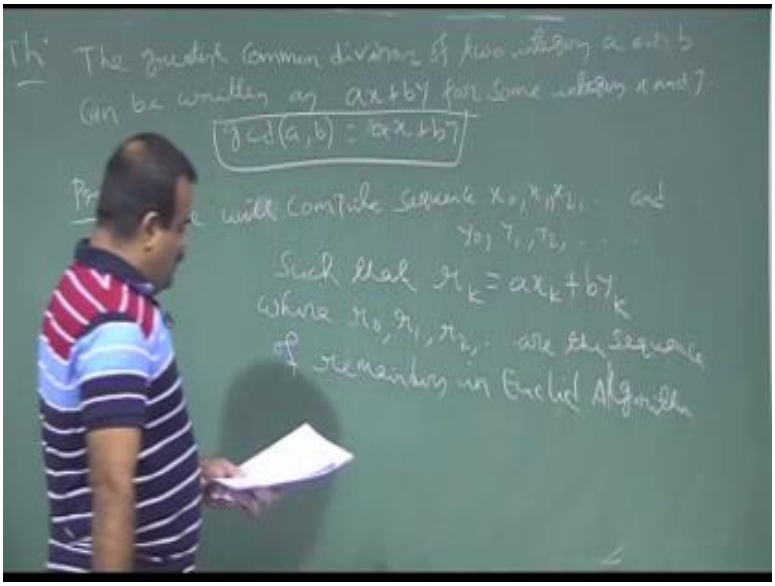
Now, to find the x y just ignore last step we got already 1, so just reverse this. So, how we can reverse this? From this equation we can write 1 is equal to 13 minus 6 into 2.

Now 2 is basically 41 so will just convert this side by only 41 and 54 and remaining are x and y, so that we will try. So, that is basically the extended Euclid algorithm. So, this 13 minus 6 into 2, so again 2 will be written as 41 minus 3 into 13; so 41 minus 3 into 13, if you multiply this, so this is 18 and 1 - 19 into 13 this is into minus 6 into 41.

Now from this equation we can write 13 as 54 minus 1 into 41. So, that will right; so 19 into 54 minus this is minus 1 into 41 minus 6 into 41. So, this is basically 19 into 54 and this is minus 19 and minus 6. This is basically minus 25 into 41. So, this is basically 54 into 19 plus 41 into minus 25. So, this is our x and this is our y. This is the example where any gcd is here 1, so 1 is written as this 54 x, this 19 is y and this is x is minus 25.

So, now we will prove this formally by a theorem, this is through an example but this will prove by a theorem. So, let us state that theorem and that theorem will give us the extended Euclidean algorithm to find the gcd, to find the inverse sorry.

(Refer Slide Time: 11:16)

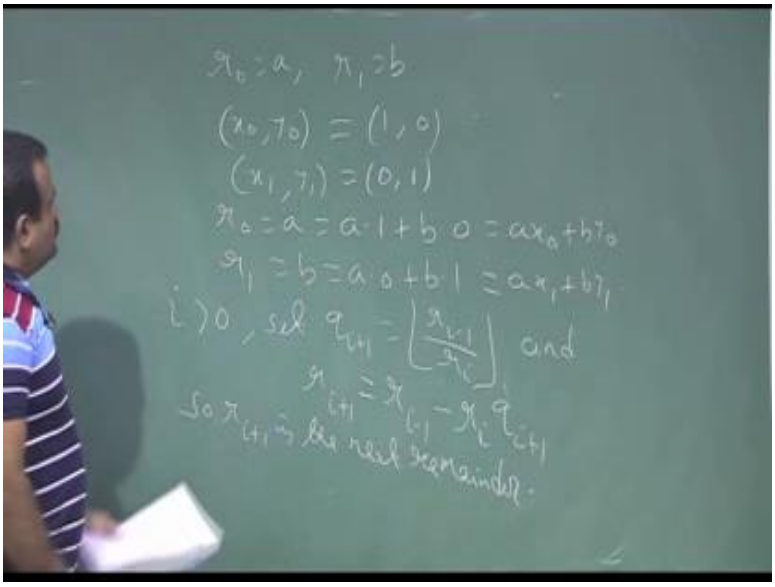


Euclid algorithm is to find the gcd, but the extended a (Refer Time: 11:14) will give us the; so this is the theorem. Theorem is telling the greatest common divisor for the gcd; the greatest common divisor of two integers a and b can be written as ax plus by for some integers x and y. That means, gcd of a b should be written as ax plus by for some x y. So, this is the theorem, we have to prove this theorem.

So, this proof is coming from the Euclid algorithm, those sequences of the remainder. So, how to prove this? Basically to prove this we will compute the sequence x 0, x 1, x 2 so on and y 0, y 1, y 2 and so on such that the remainder is basically ax k plus by k. So, these are coming from Euclid algorithm; basically remainder coming from Euclid algorithm where this r 0, r 1, r 2 are the sequence of remainders in coming from Euclid algorithm.

So, in Euclid algorithm we start with a and b.

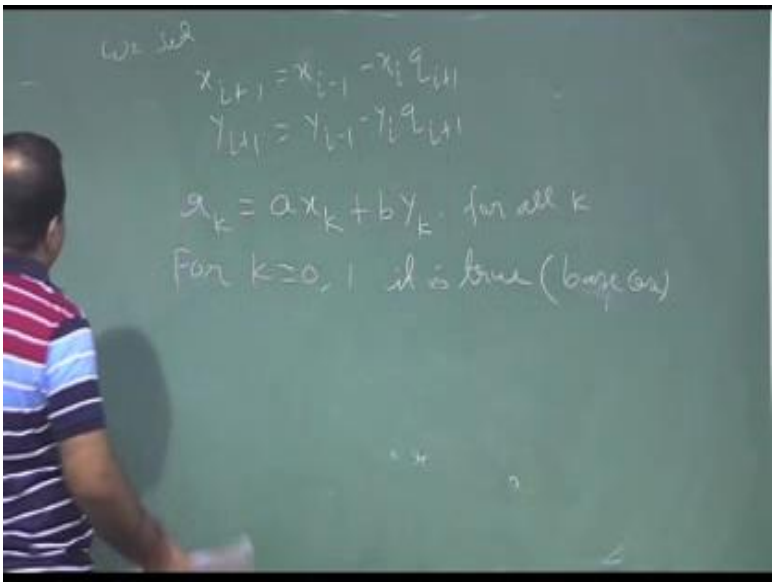
(Refer Slide Time: 14:11)



So basically what we have? We have r 0 is equal to a and r 1 equal to b this is the initial case and x 0 y 0 is basically 1 into 0 and x 1 y 1 is basically 0 into 1. That means, we can just check it. So, r 0 is basically a which is basically a into 1 plus b into 0, so this is basically ax 0 plus by 0. This is the initial case will prove by induction r 1 is equal to b which is basically written as a into 0 plus b into 1 which is our a into x 1 plus b into y 1.

And for other than 0 r 1 for i greater than 0 what we do, we set q i plus 1 that cosine is basically r and i minus 1 by r i. So, this is the division step and r i plus 1 is basically r i minus 1 minus r i into q i plus 1. So, the r i plus 1 is the next remainder. So, now what we said from this general case?

(Refer Slide Time: 16:14)

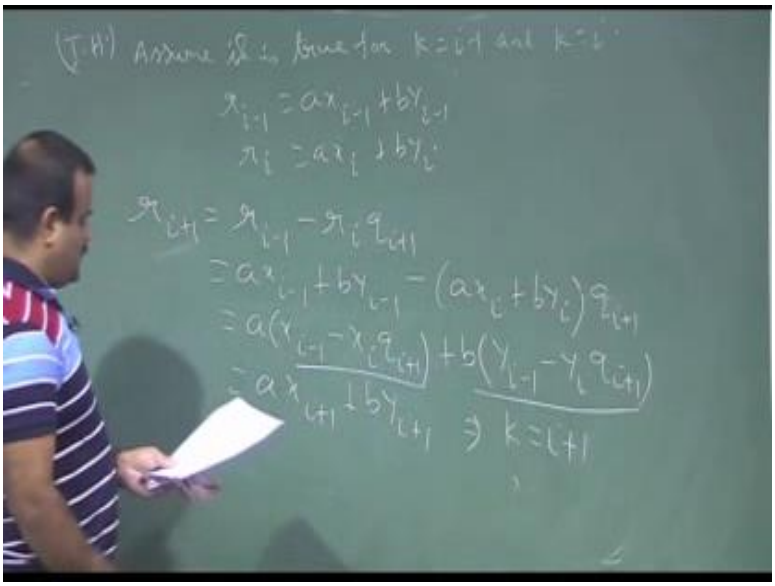


We said we have to set x i and y i. So, we set x i plus 1 is basically y i minus 1 minus x i q i plus 1 and y i plus 1 is basically y sorry, this is x x i minus 1 and then this and y i minus 1 minus y i q i plus 1. Now, we have to show by mathematical induction that r k is basically ax k for such k plus by k for all integer k; for all k. So, these we have to prove by induction. Now this for k is equal to 0 and 1 this is true because that was the base case. So now we will prove this by induction.

Now, for k is equal to 0 and 1 it is true and this is the base case, so for mathematical induction is having base case. So, we need to prove that some properties with n natural, number n so for that we prove that the statement is true for n is equal to 0 or m is equal to 1 and then we show that if the statement is true for n is equal to k then we show that statement is true for n is equal to k plus 1. Then the statement is true for all natural numbers. So, that is the one that the method of induction hypothesis induction method.

So now we assume the statement is true for k is equal to i minus 1 and i and then will show the statement is true for k is equal to i plus 1.

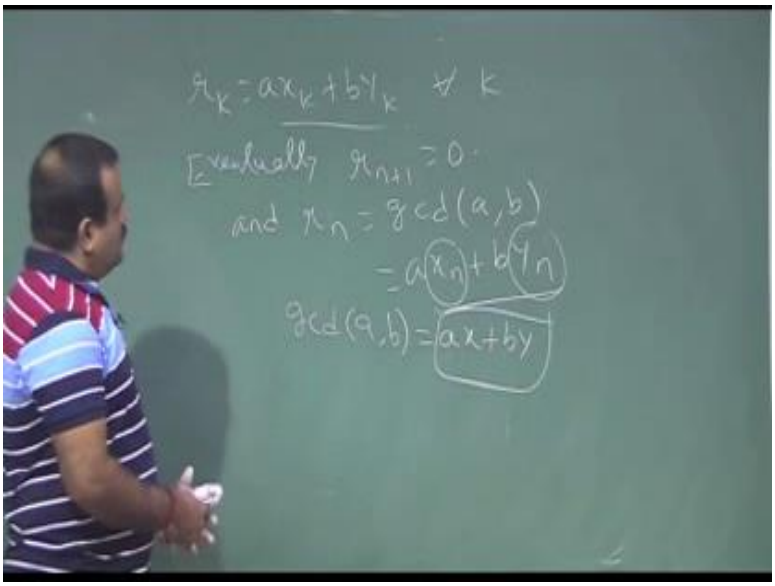
(Refer Slide Time: 18:29)



So, this is the induction cipher text is or the assumption; what is the assumption? We assume it is true for k is equal to i minus 1 and k is equal to i. That means, r i minus 1 is basically ax minus 1 plus by minus 1 and r i is basically ax i plus by i. And we have to prove that the statement is true for k is equal to i plus 1. So, for k is equal to i plus 1 what is r i plus 1, r i plus 1 is basically r i minus 1 minus r i q i plus 1. So now we put the value over here, so this is basically ax i minus 1 plus by i minus 1 minus r i is this one so ax i plus d y i into q i plus 1. So, these we can write as like this ax i minus 1 minus x i q i plus 1 plus b into y i minus 1 minus y i q i plus 1. Now we know this is basically x i plus 1 and this is basically y i plus 1; plus y i plus 1.

So, it is true for; so this implies the statement is true for k is equal to i plus 1. Hence, the statement is true for all k.

(Refer Slide Time: 20:36)

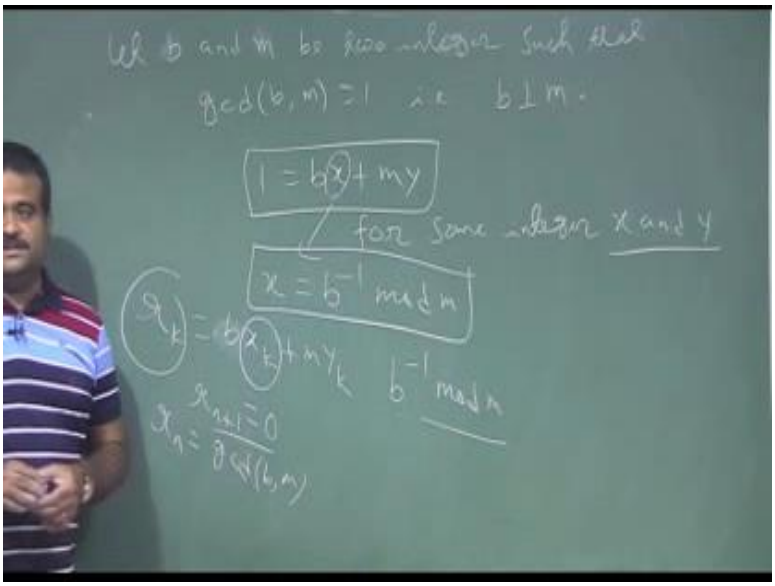


So, by the mathematical induction that means this implies r k is equal to ax k plus by k for all k; for all natural number k. And so this r i's are basically the remainder. And by Euclid algorithm eventually we get when we stop when we get the remainder is 0. So, eventually in the Euclid algorithm eventually we reach r n plus 1 is equal to 0 and then the previous r n is the gcd basically.

And r n is basically gcd of a b which is basically ax n plus by n. So, we are running the Euclid algorithm. So, it will stop when the remainder is 0 then the previous reminder will give us the gcd, and that remainder is always in this form. So, we will maintain this x k y k. So, finally this is x this is y. So, gcd will be written as ax plus by, basically it is x n y n.

So, this is the proof of that any gcd can be written as ax plus by. So, this will give us the extended Euclidean algorithm which we just every time we maintain this x k and y k. So, finally, we have this x y. Once we have x y, so if the gcd of; so will come how to get x y by Euclidean algorithm so just we follow these steps will have algorithm form we will see that.

(Refer Slide Time: 22:49)



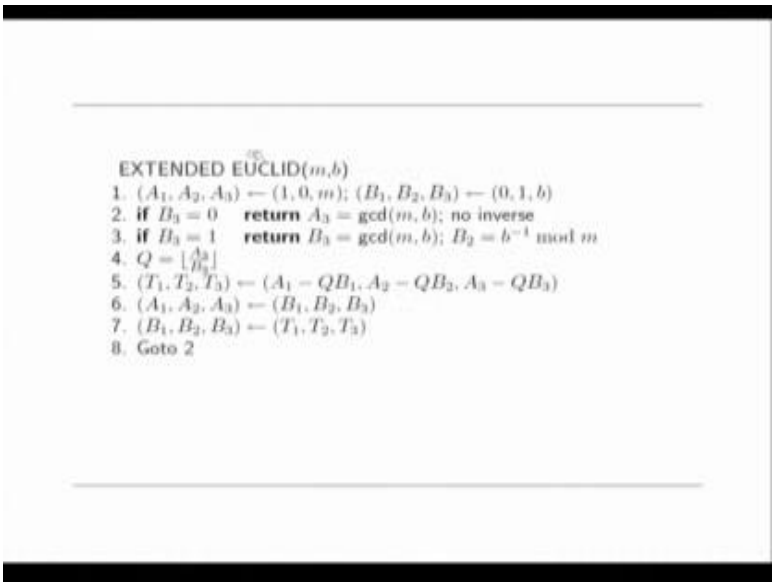
Now if suppose you have two integer let a and b be two integer say b and m; b and m be two integers such that gcd of b and m is 1 so that is they are relatively prime, so b is relatively prime again.

That means, we have proved the theorem - that means there should exist x y such that 1 is equal to bx plus my, for some integer x and y. So, this x y is basically coming from the when we run the Euclid algorithm. So, every time we write the remainder in x k a this bx k plus my k so finally eventually it will end up with the remainder 0 r n plus 1 is 0 then the previous remainder is the basically the gcd.

So now we have this form. Now from here we know this then x is basically b inverse mod n, so this is the multiplicative inverse. So, if we can find this form x y, if we can find ax then ax is basically the inverse multiplicative inverse of b under modulo m; so that we will get from the Euclid algorithm.

So, let us talk about this Euclid algorithm.

(Refer Slide Time: 24:55)

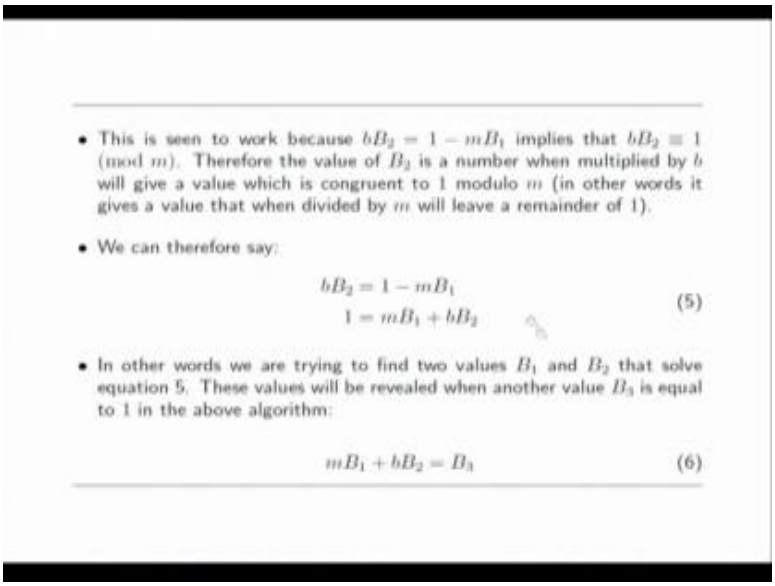


So here is the algo partial of Euclid algorithm; so basically how we have the theorem proved this is the steps basically. We want to find the b inverse mod m, if inverse exist provided gcd of b and m is 1. So, provided b is relatively prime with m then the b inverse mod m exists. So, this is B 3; B 3 basically give us the gcd.

So, what we are doing? Initially step we are taking help of some temporary variable, this Ai's we are assigning 1 0 m and B is we are assigning 0 1 b and we run this until B 3 is either 0 or 1. If B 3 is 1, that means this is the gcd is 1 otherwise we return knowing. Then B 2 will give us the inverse. So, will see this B 2, B 3, and we have a temporary variable q which is the basically quotient which is the A 3 by B 3. And this is also three temporary variable we are taking this value A 1 minus Q into B 1, A 2 minus Q into B 2, A 3 minus Q into B 3.

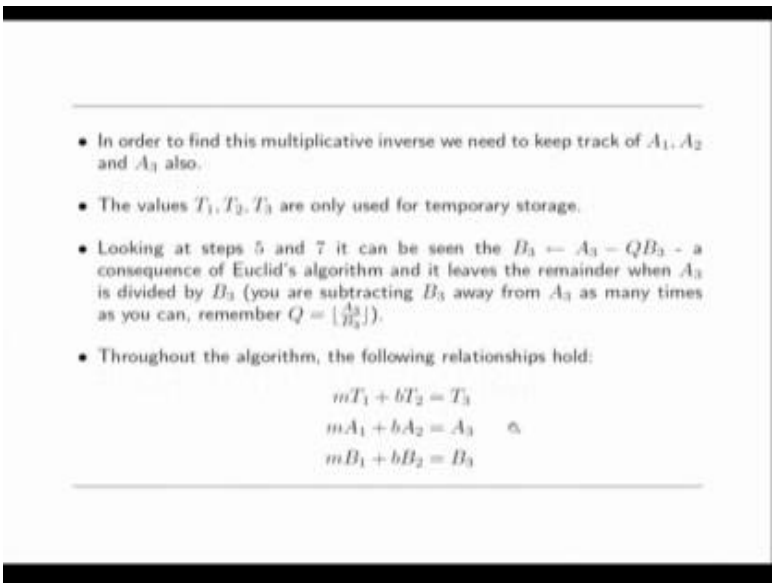
And then we copy this A 1, A 2, A 3 there this B 1, B 2, B 3 and B 1, B 2, B 3 is now replaced by T 1, T 2, T 3; the new value of this temporary variable. Then again we go to two, so until the B 3 will become 0 or 1 will continue these steps, this loop. So, this is the code for finding the gcd basically. Once we got the gcd is 1; that means, B 3 is 1 then the B 2 will give us the inverse. So, B 2 is basically the x, so we are maintaining x over here. So, this B 2 will give us the x this x. So finally, this is B 3. So, if it is 1 then gcd like this so this is B 2.

(Refer Slide Time: 27:17)



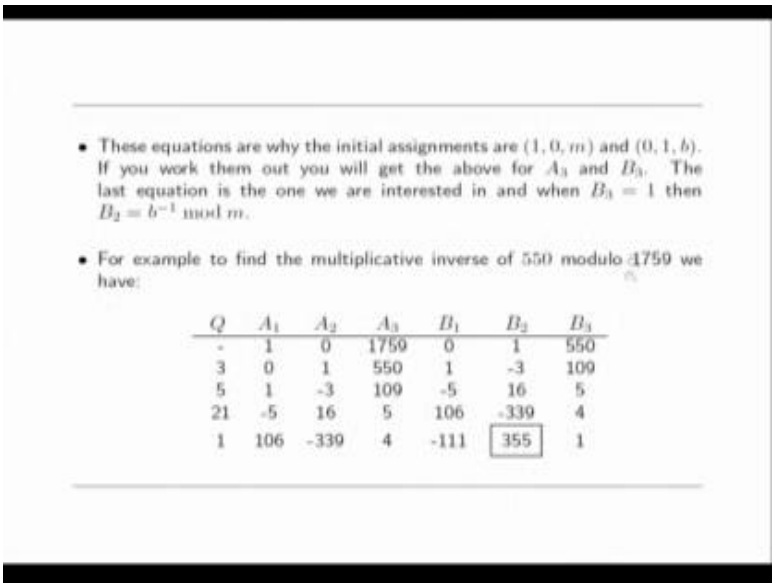
So now, this will work because it is every time we are maintaining this we are trying to store, this is the every time it is b, small b into B 2 is 1 minus m B 1. So, b B 2 is congruent to 1 mod m and every time we are maintaining this B 2 is equal to 1 minus m B 1 so 1 is equal to this. In other word we are trying to find the B 1, B 2 that solves the equation this 5.

(Refer Slide Time: 27:52)



And so this is the temporary variable and throughout the algorithm this T 1, T 2, T 3 at the temporary variable, and this is the way we are replacing B 1, B 2, B 3. And throughout the algorithm this relationship holds.

(Refer Slide Time: 28:12)



So basically, we are trying to get x and y. So finally, if B 3 is basically 1; that means, gcd is 1. Then the B 2 will give us the inverse. So, here is an example you want to find suppose this is b; small b so you want to find the multiplicative inverse of 550 modulus 1759, so this is basically our m; small m. So, what we are storing this value like our algorithm.

So, this value is basically A 1; A 1 is assign 1, A 2 is assign 0 and A 3 is assign basically m and B 1 is assign 0, B 2 is assign 1 and B 3 is basically assign b. This is the initialization step. And Q is basically A 3 by B 3 every time, so Q is 3. And then we are just changing this value by this temporary variable calculating temporary available A 1 minus Q B 1, A 2 minus Q B 2, A 3 minus Q B 3 and they will be replaced been B 1, B 2, B 3. So, this will where we continue until this B 3 will become 0 or 1. If the B 3 is becomes 1 then the gcd is 1 and the corresponding B 2 is basically inverse of that b inverse mod m.

So, this is just the execution of this code this will give us the first. Basically we are doing the Euclid algorithm and finally we are getting the gcd. Now if the gcd is 1 and every time we are maintaining this x and y. So, x and y because that r k is basically ax k; a is

different here bx k plus my k. So, every time we are calculating this r k. Now eventually this r n plus 1 is 0, so that means r n will give us the gcd of b and m. So, if r n is 1 then the gcd is 1 and the corresponding this is B 2 you are storing this value then the corresponding this value is basically b inverse mod m provided gcd is 1.

So, this is basically we are executing the Euclid algorithm, but every time we are storing this x y so that we can get the inverse of b; provided the gcd is 1.

Thank you.