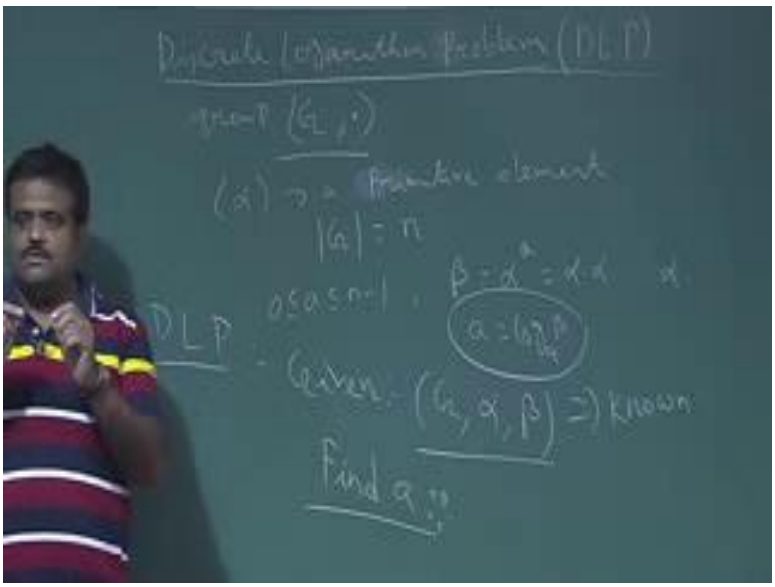
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology Kharagpur

Lecture - 57
Discrete Logarithm Problem

(Refer Slide Time: 00:20)



Discrete log problem - so, discrete log problem is basically used in we have seen the Diffie-Hellman key exchange protocol, we have seen Elgamal cryptosystem. So, there we have seen the discrete log problem. So, what is a basically discrete log problem? We have given a group G which is basically multiplicative group and this is a cyclic group and a is a say alpha is a generator of this group; alpha is a generator or the primitive elements of this group. So, it is a cyclic group generated by alpha; primitive element of this group and the discrete log problem is basically.

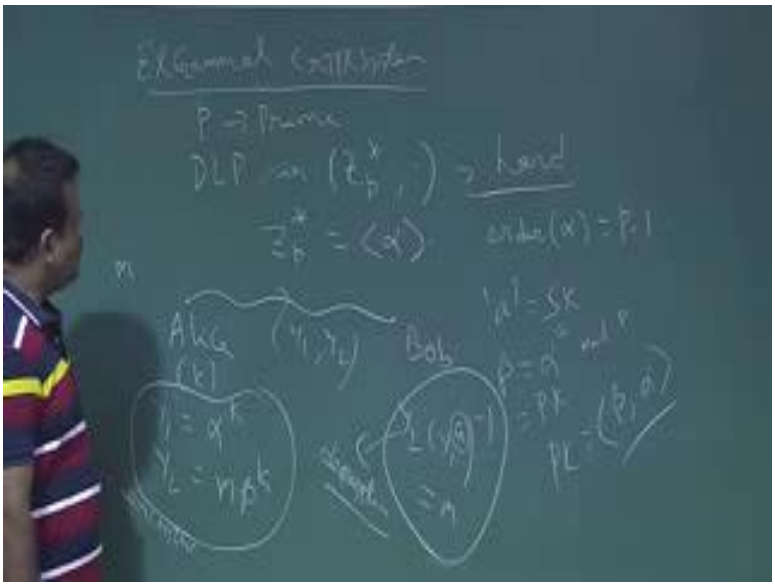
We choose this, suppose the order of G is n suppose the order of G is n so; that means, we choose a from lies between 0 to n minus 1 and we compute beta which is basically alpha to the power a. So, if it is multiplicative sense then alpha into alpha a times or it could be additive sense also. So, depending on the operator we are using. So, we compute b is equal to alpha to the power a.

Now the discrete log problem is basically what? Now given this given G alpha beta so, with these are known these are known given G alpha we said beta. So, what we need to

calculate we need to get a. So, that is the problem. So, find a, so a is usually denoted by log of beta alpha like this. So, this is the discrete log sense. So, find a, so, this problem is called discrete log problem. So, given the instance G, G is the group G must be known alpha beta alpha is the generator of the group or primitive element of the group and beta is the alpha to the power a. So, given the value of beta we need to calculate a. So, that is the discrete log problem.

So, usually so based on this discrete log problem, we know the Elgamal cryptosystem. So, basically for Elgamal cryptosystem what we have G is basically Z p. So, what is the Elgamal cryptosystem?

(Refer Slide Time: 03:27)



Elgamal is basically we have Elgamal cryptosystem. So, basically p is a prime we are choosing p to be prime such that discrete log problem is hard problem in Z p star is hard is hard means injected it. So, there is no I mean it should be computationally hard.

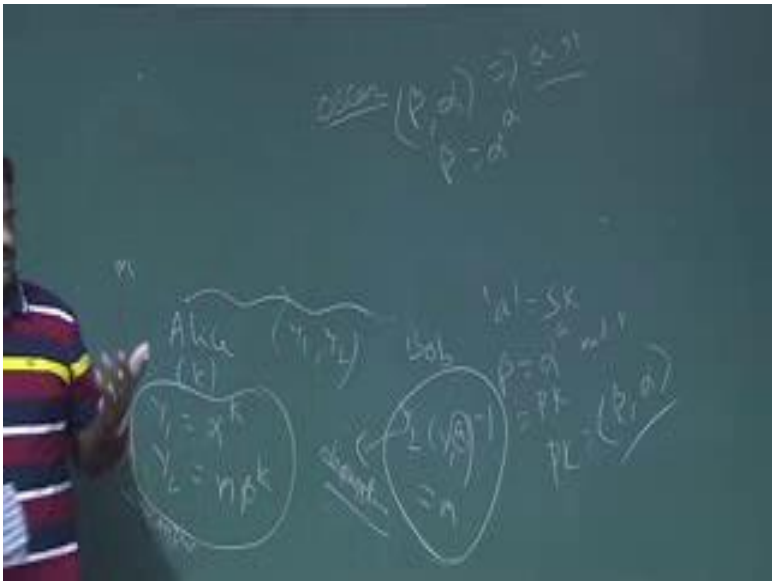
Then say alpha is generator set. So, Z p is basically Z p star is basically so, generating by alpha where alpha is primitive element and say order of the group is a order of the group is p minus 1 now what are the keys. So, Alice want to Alice is communicating with Bob. So, this key generation algorithm, so the Alice wants to send the message to Bob. So, this key generation is done by the Bob. So, what Bob is doing? So, Bob is having this group Z p star and then Bob is choosing a. So, Bob is choosing a which is basically

secret key of Bob and Bob is computing beta which is basically alpha to the power a mod p. So, now, this is basically this is for this is secret and this is the public.

So, the public key of Bob consists of beta and alpha and; obviously, p. So, the group must be known and then the secret key is a. So, what Alice will do Alice wants to encrypt message which is basically say m. So, Alice will generate the cipher text which has 2 part y 1 and y 2. So, basically Alice will choose a k random number and then Alice will compute y 1 equal to alpha to the power k and y 2 is equal to n into beta to the power k and then this is the encryption this is the encryption and what is the description? What Bob is doing? The description is basically Bob is just calculating y 2 into y 1 to the power a minus this. So, this should give us M. So, this is the description this is the description which is done by Bob.

Now the thing is if so to decrypt this, Bob is using Bob's secret key. So, this is the secret to the Bob, now suppose if Oscar is knowing this a then Oscar also can decrypt it. So, the Oscar data card so, this is the public key.

(Refer Slide Time: 06:51)



Oscar knows beta Oscar knows alpha. So, if someone Oscar knows a then Oscar can decrypt it so; that means, to knowing a Oscar has to so, beta is basically alpha to the power a. So, given this 2 finding a is basically hard. So, this is basically discrete log problem is hard. So, a security or Elgamal cryptosystem is based on the discrete log problem hard.

Now we talk about how to what is what is the algorithm to finding this the handle this problem discrete log problem. So, let us start with the exhaustive search for this problem. So, we will now talk about solving the discrete log problem.

(Refer Slide Time: 07:51)

The first approach we will do the exhaustive search exhaustive search or boot post method.

What is the problem? So, we have given a we have a group G which is basically generated by alpha and the order of this group is say p and we have given a beta which is basically alpha to the power a where a is for p minus 1 and we need to. So, this is known this 2 is known and we need to this 2 is known and from here we need to find this a. So, that is the problem that is the discrete log problem. So, how to find a?

What is the exhaustive search method or exhaustive search? So, we know the - a is from 0 to p minus 1. So, what we do? We will try for all possible a. So, we first choose some a. So, a is equal to 0 or a is equal to 1. So, we just do like this. So, alpha square alpha cube we keep on calculate this alpha to the power i like this and every time we try to match this with whether this is equal to beta or not question mark whether this is equal to beta or not question mark.

If this is beta then our k is, our a is basically i. So, this is the exhaustive search or brute force attack. So, we keep on calculate alpha square alpha cube like this and every time

we check that alpha to the power k is beta or not if suppose alpha to the power k is happened to be beta if; that means, that k is our a. So, that is the exhaustive search, but these will take how many times this will take order of order of if this is p order of p times i mean.

Now, how we can be fast is, so this is a, we can make it faster by doing the table lookup. So, what we do? We just if you have if you are allowed to do some p possessing. So, what we do? So, this is called table look up; table look up methods.

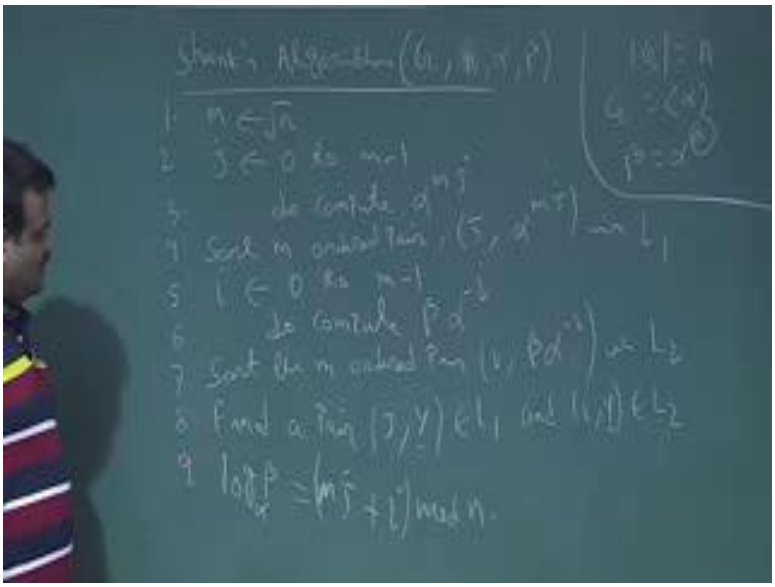
(Refer Slide Time: 10:27)

What we do? We store this pair i alpha to the power i. So, for all possible i, so, i is equal to where 0 1 2 up to p minus 1. So, we store this in a table with respect to we sort this basically sorted with respect to the end point this is the start point this is the end point with respect to the end points .

We can use any sorting algorithm to do that where if it is sort it and we store which maintain a table where we store this. So, now, then we have given a beta, beta is basically alpha to the power a now we are given beta. So, what I do? We sort this beta in to the table. So, these are sorted now we can search using the binary search. So, since these are sorted this sort will take how much time logarithm time. So, we search it now suppose it is matching with alpha to the power k now this k is basically our a.

If alpha to the power k is happen to be beta this imply a is k. So, this is the table look up method then this will take the memory also because we are going to store these all the tables in a array like this. So, another method is Shank's algorithm.

(Refer Slide Time: 12:18)

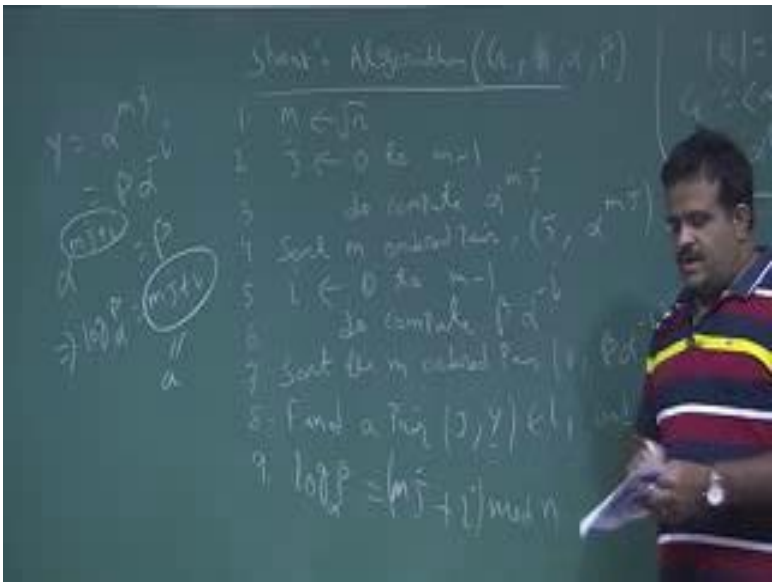


These are this is another method to finding the discrete log problem Shank's algorithm. So, this is also taking G n alpha beta or G p, say p is the order of the group if you take n is the order of the group. So, order of the group is n and alpha is the primitive element for the group and beta is basically alpha to the power a and we need to find this a. So, this is the problem.

This algorithm is telling is like this. So, in first row M is equal to root over of n and then we compute where j is equal to 0 to m minus 1, do we compute this value? We compute alpha to the power m into j. So, alpha to the power m to the power j so alpha to the power m into j and then we sort this. So, there are m ordered pair we sort this m ordered pair they are basically j comma alpha to the power m j in a list L 1. So, may be in the 1 sort with respect to the end point in the list L 1 now also it appear a list another list L 2 like this. So, this is for i is equal to 0 to m minus 1 we compute we compute this value beta into alpha to the power minus i and these we store into a list L 2 in a sorted we sort the m ordered pair this m ordered pair so, i comma beta alpha to the power minus i in L 2.

Here 2 ordered pair, now we have 2 list where we are sorting this and this, now we match the end points and suppose we find the pair. So, this is 8, suppose we find the pair in the L 1 and L 2 also such that then M point is matching suppose we find a pair say j comma y in L 1 and i comma y and L 2 having the same y value this same end points then that will give as then over this log of beta alpha is basically m j plus i mod n m j plus i mod n. So, why this is? So, because so, we know y is equal to from the list 1.

(Refer Slide Time: 16:06)



We know from n 1 we know y is equal to basically alpha to the power m j. So, if it is same as beta to the power beta into alpha to the power minus i. So, if this is also y then basically what we have we take this alpha this side we have alpha to the power M j plus i is equal to beta. This implies this is our log of that. So, these imply log of beta alpha is equal to M j plus i. So, this is our a, we are looking for. So, this is the correctness. So, we will take out quick example on this. So, let us a take a quick example on this. So, suppose we want to calculate.

(Refer Slide Time: 17:05)

Suppose we want to calculate log of log 3 is 525 in Z 809 and this is a multiplicative group.

Now, 809 is a prime. So, this is our p and 3 is a primitive element in Z p primitive element in Z 809 star. So, here alpha is three n is 808 basically p minus 1 and beta is basically 525. So, we need to find out a. So, we will just run that Shank's algorithm. So, for that we take the n which is basically a square root of M 808 upper cement. So, this will give us the value 29. So, we compute alpha to the power M because you have to again compute alpha to the power M to the power j. So, we compute alpha to the power M so alpha to the power 29 which is mod 809 which will give us 19.

Now we first prepare the list L 1. So, to prepare the list L 1, we have to calculate this j, j M alpha to the power alpha to the power m into j basically. So, basically 99 to the power j mod 809, so, this is the values which will be stored in L 1.

(Refer Slide Time: 19:17)

If you do that this is our L 1. So, let us start the L 1 and this is for j is equal to j is from 0 less than or equal to j less than equal to 28. So, if you just calculate this, let us quickly write that. So, 199 293, so we can be just easily check this 3084 comma 559 then 5 comma 329 and basically dot, dot, dot, 10 comma 644 then 11 comma 6654 anyway one must calculate this dot, dot, dot and the last one is 28 comma 81. So, this is the list L 1.

Similarly now we have to calculate the list L 2. So, how to get L 2, to calculate L 2 we need to d bar beta into alpha to the power minus 1.

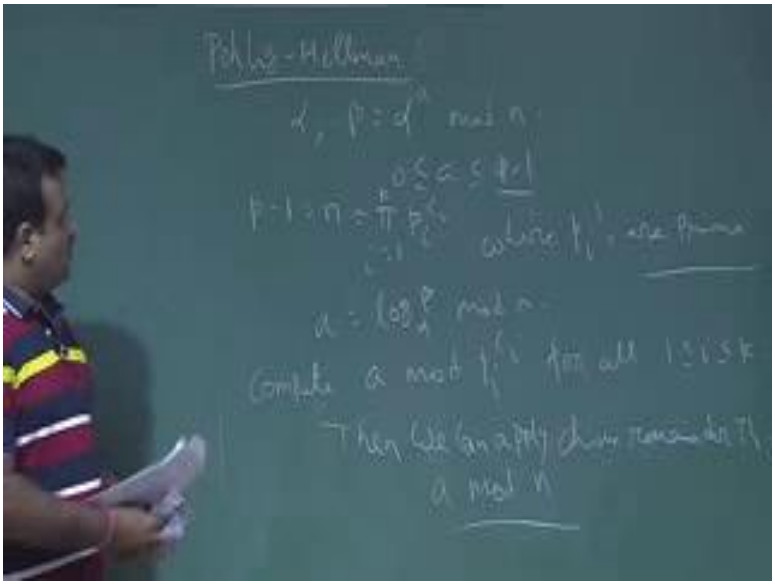
(Refer Slide Time: 20:42)

To calculate L 2 the second list we have to calculate i comma 525 into 3 to the power i minus 1 and this i is this is this mod 809 and this i is valid for 8. So, if you do so then you will be getting like this, this is the L 2 list 0 comma 5251 comma 752 comma 328. So, like this if we continue then 9 comma 511 dot, dot, dot, then we have 19 comma 644, this is 19 comma 644 dot, dot, dot and the last one is 28 comma 163, these 2 list. So, these 2 lists are sorted based on the L 1 and L 2. So, among these 2 lists we just try to get the match the end points. So, this is the matching here this is our y.

If this is our y then; that means, this is our corresponding i and j and this is our i. So, hence we get log of this 525 basic, 3 is basically 29 into 10 plus 19 mod 808. So, this is basically 309. So, this is basically our result, we will be keep for we can verify this by just calculating three to the power 309 is basically congruent to 512 mod 809. So, this is the science method.

Now, we will talk about another method which is called Pohlig-Hellman so basically.

(Refer Slide Time: 23:25)

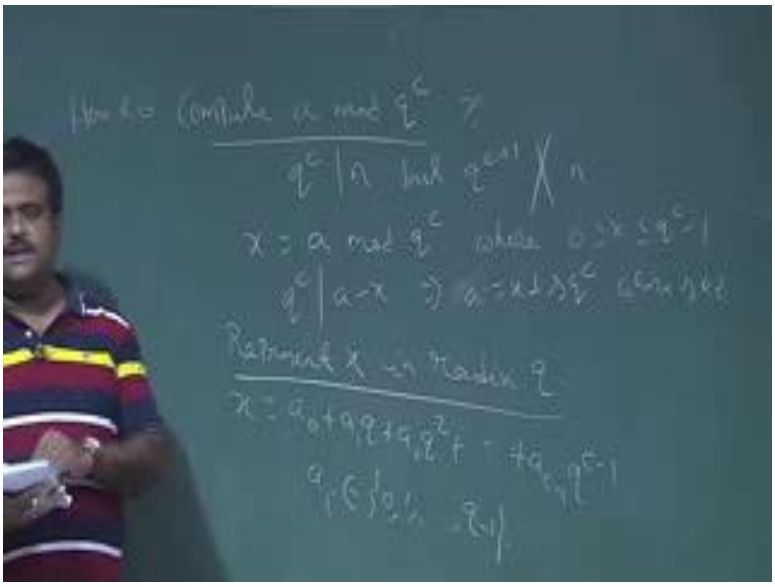


Pohlig-Hellman method for solving the discrete log problem so basically the idea is so, we know that our alpha we need to find out alpha a such that a to b or beta is a to the power alpha mod n now what is the order of alpha order of alpha is n and a is between 0 to n minus 1 now we choose this 0 to p minus 1 basically and we choose this p minus 1 as n and p is prime. So, p minus 1 is composite. So, if composite it can be written as product of prime. So, that is a fundamental theorem of arithmetic. So, in must we are

able to write this as a product of prime p to the power i c to the power i where i is 1 to k. So, where k i's are prime number.

Now our objective is to find the - a, a is basically log of beta these alpha mod of n. So, now, if we can compute if we can compute a mod p to the power i c to the power i for all of this i for all i then one can apply the Chinese remainder theorem then we can apply Chinese remainder theorem to get the -a mod n. So, that is the idea. So, to get an is a if we can identify if we can find a mod p to the power p i to the power c i then we can by apply Chinese remainder theorem we can get this is the now the question is how to get a mod p to the power c. So, that we have to find out.

(Refer Slide Time: 26:02)



Now the question is how to compute a mod say q to the power c where q is a prime now q to the power c how to compute. So, this is a question mark now q to the power c divisible by n that q to the power c plus 1 is not divisible by n. So, now, to determine x is equal to a mod q to the power c where x is basically lies between q to the power c minus 1 so; that means, q to the power c must divides a minus x. So, these imply x is basically x is basically so this imply a is basically x plus s q to the power c where s is an integer now we write x in terms of the radix q represent x is radix q.

So, how we can do that? So, we write x to the power as it is radix q i mean you know the octal system you know the decimal system. So, this is basically a 0 plus a 1 q plus a 2 q square like this. So, dot, dot, dot, a c minus 1 q to the power c minus 1, so, then basically

a is basically x plus s q to the power c now the question is how we can find this a i's though these a i's are unknown. So, a i's are unknown basically so, 1 2 up to q minus 1, these are unknown now the question is how we can how we can determine these a i's.

To determine this a i, we need to use this we want to use this result like.

(Refer Slide Time: 28:22)

Now, first of all, how to compute a 0? So, to compute a 0 what we do we claim this result claim this beta to the power n by q is equal to alpha to the power a 0 n by q. So, this e can be easily verified. So, basically beta to the power n by q is basically alpha to the power a n by q. So, now, this is basically alpha to the power summation of a i q to the power i plus s q to the power c n by q. So, this will, basically all terms will vanish mod n. So, we will all the late a 0 n by k.

Now if you take this gamma is equal to alpha to the power n by q, now we keep on calculating gamma square gamma cube until gamma to the power i is equal to beta to the power n by q and this is a exhaustive search for some i and that i is basically our a 0. So, if c c 1 is 1. So, this is for all i and that i is basically our - a 0. So, now, once we have a 0 we will keep on calculating we will try to get a 2 a 3 like these. So, this is the way we try to get all the - a's. So, once we have all a's then we will get the - a to the power basically we got the x a, a mod q to the power c. So, once we have the - a mod q to the power c then we can have, we can apply the Chinese remainder theorem to get the - a mod n. So, that is a general that is the form of the Pohlig-Hellman method.

Thank you.