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

Lecture - 28
ElGamal Cryptosystem

We talked about another public key cryptosystem which is called ElGamal cryptosystem, and this is based on the discrete log problem is hard.

(Refer Slide Time: 00:34)

So, let us define the discrete log problem, discrete log problem over the logarithm problem over we have seen this in the Diffie-Hellman key exchange. So, problem is let p be a prim, if p be a prime then Z p star which is basically Z p minus 0 this will give, this will be a cyclic group and then let alpha be a generator of this group, alpha be a generator of primitive element - element of Z p star. Then given this, then we compute beta which is basically beta is alpha to the power a mod p, where a is 0 less than equal to a less than p minus 1 or equal to p minus p. So, problem is discrete log problem is given the value beta, alpha and p, given this we need to compute, need to find a. So, this is the problem, this is called discrete log problem.

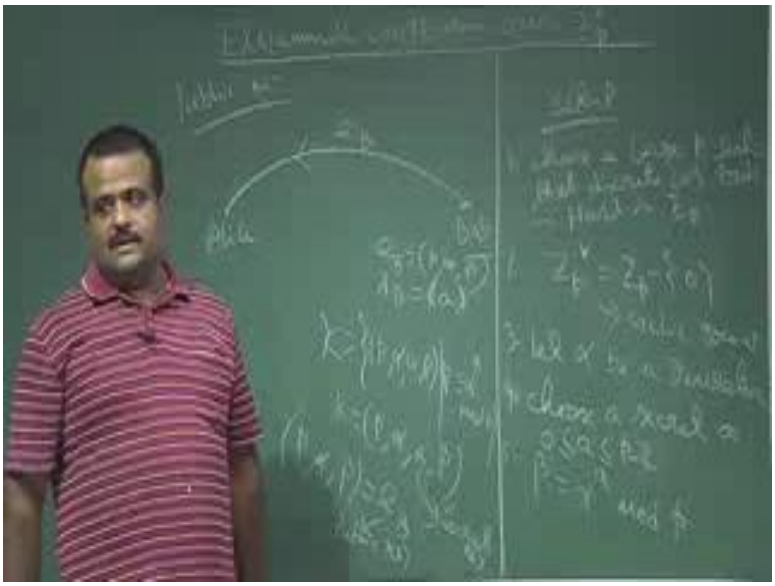
So, given beta, so this is sort of I mean, if all are integer this is sort of real number, this is sort of log of beta base a kind of thing. So, this problem is called discrete log problem, given beta, alpha, p we need to find the a. So, the boot force method is we can try for all

possible a. So, we know a is from 0 to p minus 1. So, we choose plus 0 then we compute alpha to the power 0 we check whether this is beta or not, we choose is equal to 1 like this, so this is exhaustive search or the boot force methods. So, they are time complexity for this, time we will take the time of the; I mean order of this prime number. So, if p is large then the discrete log problem is hard in this, so if p is large then the discrete log problem is hard problem over Z p star is hard; that means, which is not capable in this integral.

So, this is the discrete log problem and we have seen the defilement key exchange protocol is also based on the discrete log problem. Now this public key ElGamal is based on this discrete log problem and based on the hardness of this problem discrete log problem is hard, like RSA is based on the hardness of the factorization if we have a given a prime which we know that it is product of, if we given an integer which you know it is a product of two prime now to get that p q is hard n is equal to p q, so that is the hardness for RSA now here hardness is the - hardness for this discrete log problem.

So, let us just talk about ElGamal cryptosystem. So, this is a public key cryptosystem, so one has to generate the public key private key pair in order to receive the message ElGamal.

(Refer Slide Time: 05:09)



So, the same story Alice wants to send a message to Bob. So, Bob is the receiver Alice is the sender. So, Alice needs to get this the public key set up, where a public key set up.

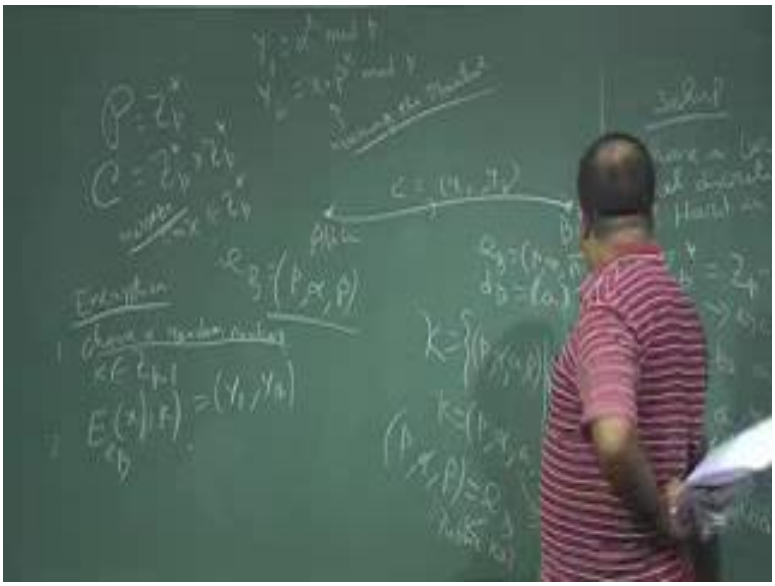
So, Alice wants to get Bob public key in order to encrypt the message. So, this set up has to done by Bob. So, Bob has to generate is public key private key pair. So, what Bob will do? Bob will first choose a large prime, choose a large prime p such that discrete log problem, this is the ElGamal cryptosystem over Z p or Z p star. So, Bob choose a large prime p such that discrete log problem is hard, is hard in Z p. So, then Z p is a Z p star is basically Z p minus 0 is a cyclic group. Now, so it has a generator.

So, Bob will choose the generator let alpha be the generator or primitive element generator or primitive element of Z p star. So, it will generate the groups. So, alpha, alpha square like this, this is cyclic group in multiplicative sense. So, now, Bob choose a secret, choose a secret a - a is lies between and Bob compute alpha to the power beta which is basically alpha to the power a mod p. So, this is the setup phase.

So, now, so after this Bob is; so, the key space is basically; the key space is basically p alpha a beta where beta is congruent to alpha to the power a mod p. So, each key is basically having this part p alpha a beta. So, among this, this is the secret key, secret key or the private key of the Bob and the remaining this p alpha beta this is basically e B. So, this is the public key of Bob, this is the public key of Bob and this a is the private key of Bob. So, Bob compute e B which is basically e alpha beta and d B which is basically a; a is the secret key of Bob. So, Bob send this e B over this public channel to Alice because Alice wants to send a message to Bob. So, Alice needs to get Bob public key in order to send a message to Bob, encrypted message to Bob, in order to encrypt the message to Bob, in order to encrypt a message for Bob.

So, this is e B which is basically p alpha beta which is the Bobs public key. Now what is the encryption in for encryption? So, for encryption what Alice is doing - on top of it Alice is choosing a secret a random number, this is Alice is doing.

(Refer Slide Time: 10:17)



So, before that what is p? P is basically Z p star the plaintext space and c is ciphertext is basically ciphertext has two part - Y 1 and Y 2, it is basically Z p start cross Z p star and key is this, key space it has four portable like this. So, the encryption is as follows this is the ElGamal encryption.

So, for encryption first Alice will choose a random number choose a random number random number k form Z p minus 1, this is the extra bit of secrecy. So, this is nothing to do with the key. So, this is a extra, this random number Alice is choosing and this is not known to Bob. So, this is maybe after choosing the message Alice choose this, this is sort of maybe some timestamp norms it called, anyway. So, now, Alice will encrypt the message x. So, x is coming from Z p star this is the message, this is the message. So, this is the encryption. So, Alice is encrypting the message using the Bob public key e B and using this k. So, k is another input this randomly chosen and it has two part, messages having two part - Y 1 and Y 2, ciphertext has two part - Y 1 on Y 2 that is why the ciphertext is the Cartesian product Z p star cross Z p star.

So, what is Y 1? Y 1 is basically we will write it here. So, this is Alice is doing, so Alice is sending this Y 1, Y 2 now what is Y 1? Y 1 is basically - alpha to the power k mod p and Y 2 is basically x in to beta to the power k mod p. So, this both the values Alice can calculate because Alice is choosing this k randomly, Alice is having alpha which is the public key part of the public key of Bob and that p is also public, x is the message Alice

is chosen and beta is public. So, Alice can compute beta to the power k mod p. So, this is the way we are masking that key this is called masking the plaintext, this is we have asking the plaintext.

So, now, this is the encryption. So, this is the ciphertext c. So, this is sending over the public channel to Bob now, how Bob will decrypt it? So, Bob is getting this message, Bob is getting this ciphertext and after receiving the ciphertext Bob has to get back the message x, so how Bob can do that?

(Refer Slide Time: 14:27)

So, that is called decryption ElGamal decryption and this has to be done by Bob, ElGamal decryption and this has to be done by Bob after receiving this Y 1, Y 2. So, Bob is receiving Y 1, Y 2. So, what Bob will do? So, decrypt of Bob has to use is d B Y 1, Y 2. So, this is basically Bob will compute this Y 2, Y 1 to the power a inverse mod p this is the decryption, this is the ElGamal decryption.

So, Bob will simply compute Y 1 Bob is receiving, Y 1 to the power a - a is the secret key of Bob basically. So, Bob can compute Y 1 to the power a then Bob will get the inverse and then Bob will multiply this with Y 2 and take the mod p and this should this suppose to give us the message x back. So, you have to verify that. So, what is the guarantee this will give us x? So, this is because see Y 1 is what? Y 1 is basically this one. So, Y 1 is alpha to the power k actually Bob is not knowing k without knowing k Bob should able to decrypt it that is the beauty of this cipher. So, Bob is; Y 1 is this mod

p; now what Bob is doing? Bob is doing Y 1 to the power a. So, which is basically alpha k to the power a mod p which is nothing but alpha to the power a to the power k mod p which is basically beta to the power k mod p.

Now, Y 2 is also what is Y 2? Y 2, so basically we are doing the Y 2 into Y 1. So, if you take the inverse of this. So, Y 1 to the power; Y 1 to the power a inverse this will give us beta k inverse mod p, so beta k inverse mod p. Now if we just put it here. So, this is basically a x into beta k and then beta k inverse mod p. So, this will give us basically x the message or the plaintext. So, this is the decryption we are getting at the message x. So, x is the message we are getting from this, so we can just write this side this may not be visible there. So, basically it is, it is basically x into beta to the power k into beta to the power k inverse mod p. So, these two are cancelling out it is basically x mod p it is basically x because x is coming from Z p star.

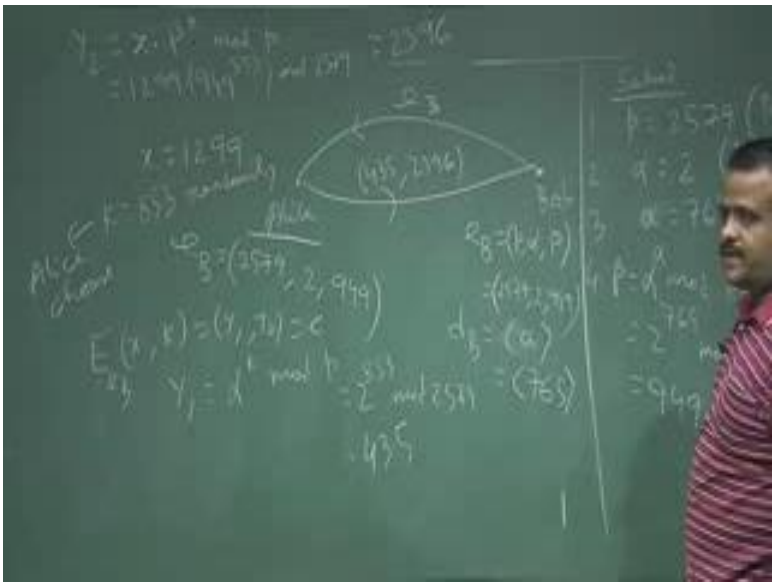
(Refer Slide Time: 17:35)

So, Bob is getting the plaintext. Now without knowing the k, without knowing this extra I mean randomness in the ciphertext given by the Alice, without knowing the k Bob can able to unmask this, by this way. So, this is the beauty of the cipher and this is not the deterministic algorithm this encryption because we are having this random k. So, this random k, for this random k many ciphertext this is not unique I mean for a given message the ciphertext is not unique, because random k is involved. So, this is not, this is

this is non deterministic because ciphertext depend on the board x and the random value k chosen by the encryptor or the sender.

So, there will be many ciphertext that are encryption of the same plaintext. So, that is the non deterministic cryptosystem, ElGamal cryptosystem is non deterministic in that sense. So, now let us take an quick example, we want to take an quick example how it is working. So, you want to take some values.

(Refer Slide Time: 19:35)



So, this is the example of ElGamal cryptosystem, cryptosystem. So, this is again we have to run the set up phase. So, Alice needs to get this plaintext and ciphertext pair. So, that Alice can; sorry Bob need to generate this plaintext and ciphertext pair. So, that Bob can make it public; Bob has to generate the public key and private key pair and can make the public key public so that Alice can send encrypt the message using the Bob public key.

So, this setup has to be done by the receiver. So, this is the set up phase. So, we need to choose a prime. So, let us choose the prime say 2579 provided this discrete log problem is hard, but this is very small prime anyway this is an example. Now it chooses a generator or the primitive element. So, two, we can check two, two generate the Z p star here. Now Bob choose a secret a, which is the secret key of Bob say this is the private key of Bob, private key. So, this is public, this is also public and this is private, a is private.

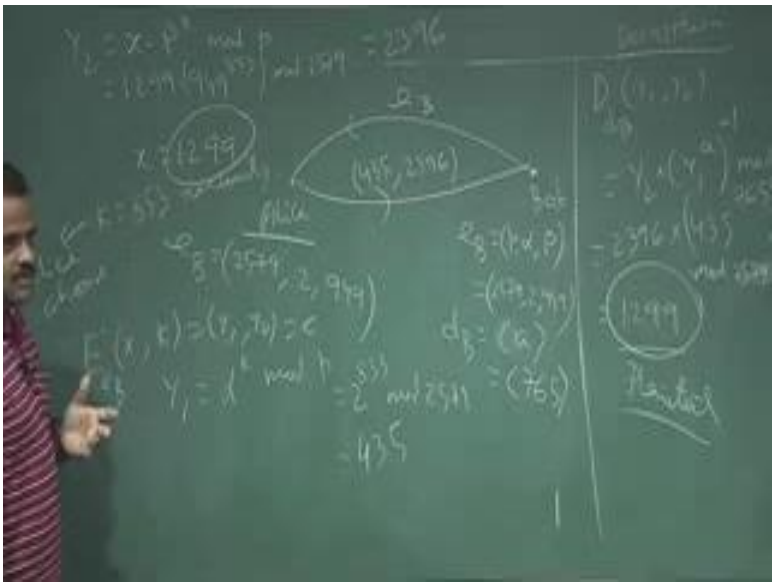
Now, we have to calculate the beta. So, beta is basically alpha to the power a mod p. So, this is basically 2 to the power 765 mod 259. So, if you calculate this, this will give us 949, so this is beta. So, now, Bob has both the public key and private key pair. So, e B, e B is basically all this component, so this, this beta is public this is also public. So, only secret is that a. So, this is basically p, p then alpha then beta this is a public key, p is basically what? P is 2579 and alpha is 2 and beta is 949 and the secret key of Bob is basically a, a Bob is chosen as, this is just a which is basically Bob is chosen by this and Bob makes this public - this e B over the public channel or in the public key directory. So, Alice is having this e B Bob public key because Alice wants to send a message to Bob. So, Bob needs to run this setup phase in order to get is public key private key pair.

So, now suppose Alice wants to send a message to Bob and the message is say 1299, this is the message Alice wants to send to Bob. So, now, Alice has to choose a random number which is k. So, suppose Alice is choosing k say 853 randomly this is Alice is chosen, Alice choose this k then Alice generate, Alice encrypt the message using the Alice encrypt this x with this k using the Bob public key and generate Y 1, Y 2 this is a ciphertext c.

So, what is Y 1? Y 1 is basically, Y 1 this basically we know alpha to the power k mod p. So, here alpha is what? Alpha is basically 2, it is 2 to the power k - k is basically this value this random number Alice is choosing 2 to the power mod p, p is basically 25. So, if you calculate this will be getting 435, 435. So, now, the question is how to reason. So, now, how to get Y 2 - Y 2 is similarly, so Y 2 is basically masking x with the beta to the power k mod p mod p. So, x is basically this one 1299 into beta, beta is basically this value 949 to the power k, k is the Alice randomly chosen value this mod p, p is basically 2579. So, if you calculate this will be getting the value Y 2 as 2396. So, this is the Y 1, Y 2 and this is the ciphertext we send it to the Bob. So, this is 435, 2396 this we send it to the Bob.

So, now, this is after receiving this ciphertext Bob has to decipher it, Bob has to get back the message. So, now, you have to run the decryption algorithm. So, this has to be done by Bob after receiving the ciphertext.

(Refer Slide Time: 26:40)



So, decryption, so for decryption what Bob will do? So, this is Bob is having Bob secret key d B which is basically a and this is Y 1, Y 2. So, Bob will calculate this Y 1 to the power a. So, Y 2 into Y 1 to the power a mod p and that will give us a x. So, what is Y 2? Y 2 is 2396 and Y 1 is 437; 435 and a is basically what? a is the Bob secret key which is 765 and then this is sorry; this inverse, inverse of this mod p - p is basically mod, this whole mod 2579, this whole mod 2579. So, this will be out of the mod 2579.

So, this we can calculate easily I mean if we calculate this inverse and all this things we should give us 1299, so 1299. So, this is the plaintext. So, this is the plaintext which was encrypted. So, see without knowing the value k which is the random value chosen by the Alice, Bob able to decrypt it. So, that is another level of secret key in this cryptosystem we can. So, this k is basically some sort of when the times star or norms. So, this makes this cryptosystem non deterministic because for different values of k we are getting the different different ciphertext for a given plaintext.

Thank you.