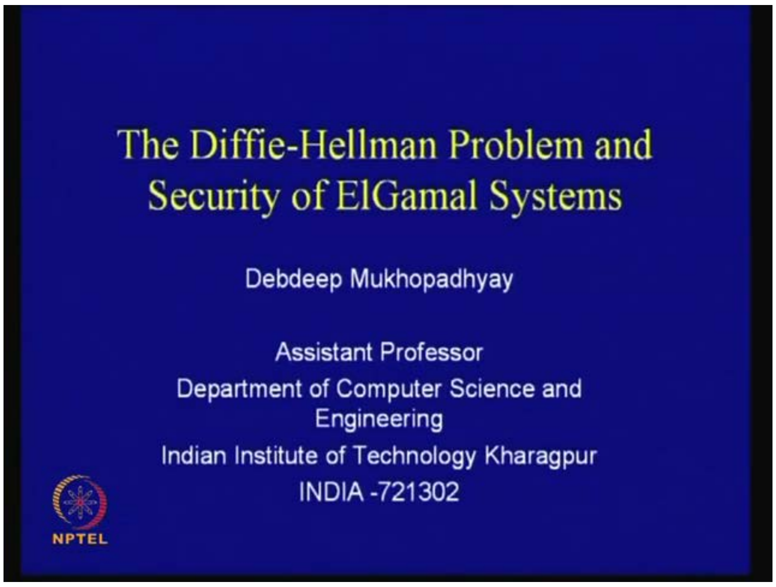
# Cryptography and Network Security

**Prof. D. Mukhopadhyay**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

**Module No. # 01**
**Lecture No. # 33**
**The Diffie-Hellman Problem and Security of ElGamal Systems**

So, in today's class, we shall discuss about the diffie-hellman problem and security of ElGamal systems. So, the objectives of today's discussion will be the diffie-hellman problem.

(Refer Slide Time: 00:21)

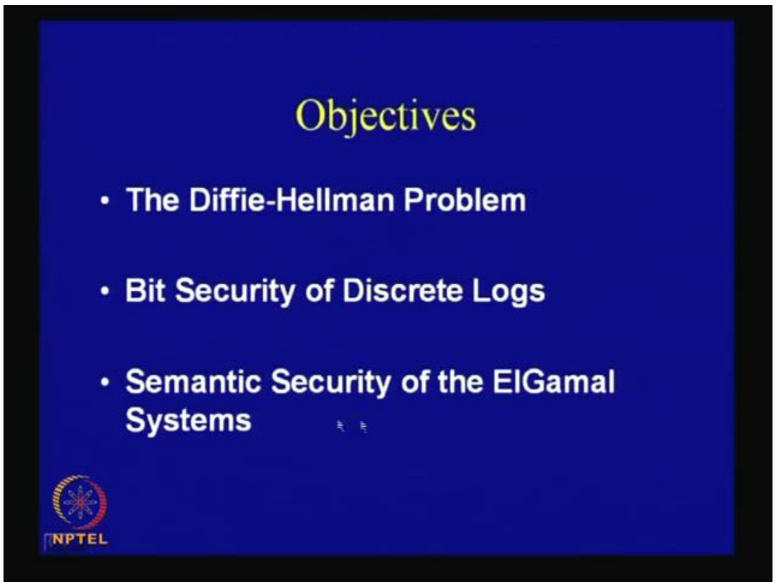


## The Diffie-Hellman Problem and Security of ElGamal Systems

Debdeep Mukhopadhyay

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
INDIA -721302

(Refer Slide Time: 00:29)

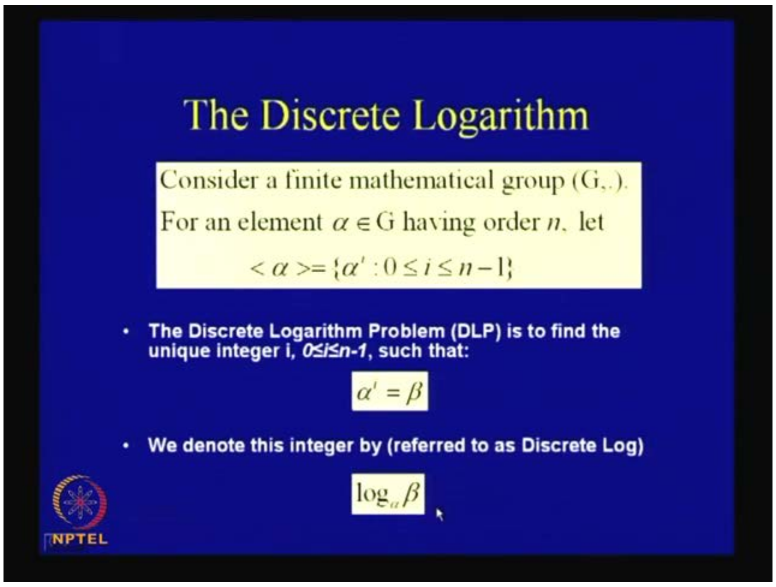


## Objectives

- The Diffie-Hellman Problem
- Bit Security of Discrete Logs
- Semantic Security of the ElGamal Systems

We shall essentially continue with the discrete logarithm problem, which we saw in the last class and we shall discuss about the bit security of this discrete logs and conclude with some comments on the semantic security of the ElGamal systems.

(Refer Slide Time: 00:41)



## The Discrete Logarithm

Consider a finite mathematical group $(G,.)$.
For an element $\alpha \in G$ having order $n$, let
$$<\alpha> = \{\alpha^i : 0 \le i \le n-1\}$$

- The Discrete Logarithm Problem (DLP) is to find the unique integer i, $0 \le i \le n-1$, such that:

$$\boxed{\alpha^i = \beta}$$

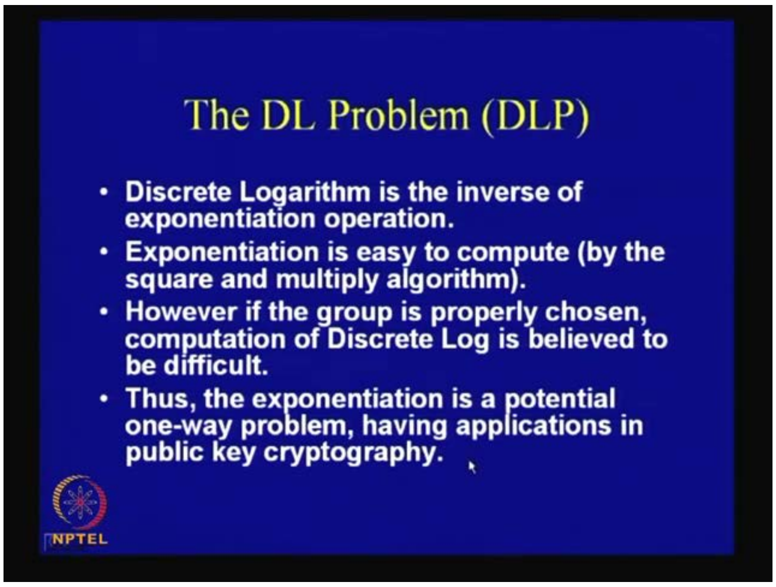
- We denote this integer by (referred to as Discrete Log)

$$\boxed{\log_\alpha \beta}$$

So, just to it is a reminder there will be as we have studied in the last class, that you know discrete logarithmic problem, we essentially have a finite group of elements and from element alpha, which belongs to g and which as an order of n, we can actually define a group by raising alpha to its various powers. So, at where like computing alpha power i, where i raises from 0 to n minus 1, we can obtain all the elements of the group. Now, we as we as we remember that, the problem of discrete logarithmic was to find out the unique integer i, when we have given alpha and beta, where alpha and beta satisfies the relation the alpha power of i is equal to beta.

So, the question is given alpha and beta you have to find the value of i. So, i is often refer to as log beta, these alpha, which is often refer to as discrete logs. So, we discussed in the last class on source on cryptanalytic methods of how to obtain these value in certain, under certain special cases.

(Refer Slide Time: 01:47)

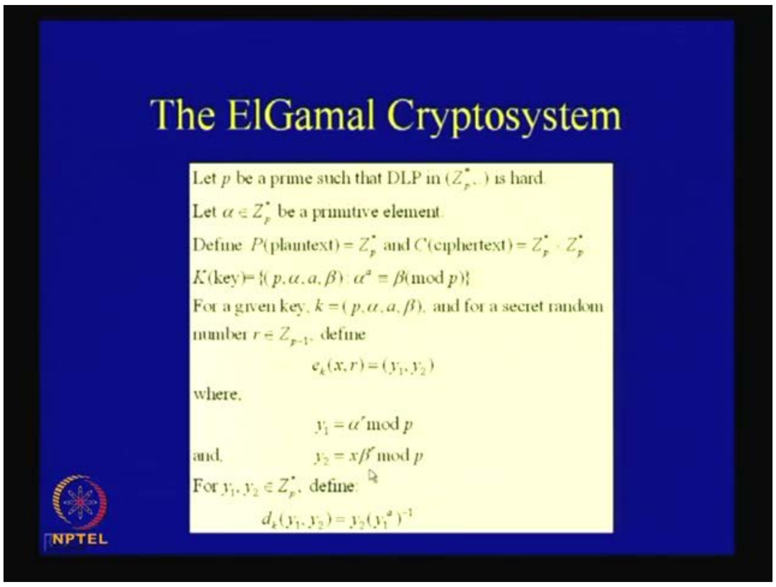


## The DL Problem (DLP)

- Discrete Logarithm is the inverse of exponentiation operation.
- Exponentiation is easy to compute (by the square and multiply algorithm).
- However if the group is properly chosen, computation of Discrete Log is believed to be difficult.
- Thus, the exponentiation is a potential one-way problem, having applications in public key cryptography.

So, the discrete logarithm problem was essentially is the inverse of the exponentiation operation and it is believe to be hard, that is we believe that for proper choices of the parameters, computation of the discrete logarithmic problem is a difficult mathematical problem.

(Refer Slide Time: 02:05)



## The ElGamal Cryptosystem

Let $p$ be a prime such that DLP in $(Z_p^*, \cdot)$ is hard.

Let $\alpha \in Z_p^*$ be a primitive element.

Define $P(\text{plaintext}) = Z_p^*$ and $C(\text{ciphertext}) = Z_p^* \cdot Z_p^*$

$K(\text{key}) = \{(p, \alpha, a, \beta) : \alpha^a \equiv \beta (\text{mod } p)\}$

For a given key, $k = (p, \alpha, a, \beta)$, and for a secret random number $r \in Z_{p-1}$, define

$$e_k(x, r) = (y_1, y_2)$$

where,

$$y_1 = \alpha^r \bmod p$$

and, $$y_2 = x\beta^r \bmod p$$

For $y_1, y_2 \in Z_p^*$, define

$$d_k(y_1, y_2) = y_2(y_1^a)^{-1}$$

Now, it has got various applications for example, we have studied the application of discrete logarithmic problems to the ElGamal cryptosystems, where essentially we have got 2 components in the cipher text. Suppose, we want to encrypt a plain text say x or and then what we do is that, we essentially choose the constants of the group. For example, we choose the prime value, we choose alpha and beta which are essentially public values, but a is a secrete, right.

So, therefore, a is a secrete and a is the logarithmic of beta with respect to alpha, that is a discrete logarithm. So, the difficulty is that, if you have given alpha, beta and also the choice and also the value of p, it this believed, it is assume that computation of a is a difficult problem. Now, based upon this assumption in this encryption scheme, what we have seen is that, if you want to encrypt for example, x where x is a plaintext value, which belongs to z p star.

Then what we do is that, we essentially choose a random number r, which belongs to z p minus 1 and then what we do is that, we choose this r, which is a random number and compute using x and r 2 parts of the cipher text, one part is y 1 the other part is y 2. Now, essentially in y 2 what we do is that, we actually blind or mask the value of x by multiplying it with beta raise to the power of r. And in the other part of the cipher text, we actually compute alpha power of r modulo p right.

So, therefore, these two things are actually transfer to the receiver, who has the knowledge of the value of a. Now, if it has got the knowledge of a, then what it does is that, it raises the part of that is y 1 the first part 2 a, which is a secrete and computes the value of alpha power of a r or alpha power of a r is nothing but, beta power of r. So, that means, the receiver can now compute the inverse of beta power of r and multiply it with y and which we back the value of x. Now, it is believe that, the person or the attacker, who does not have the knowledge of a, right cannot compute this inverse because that is a challenge.

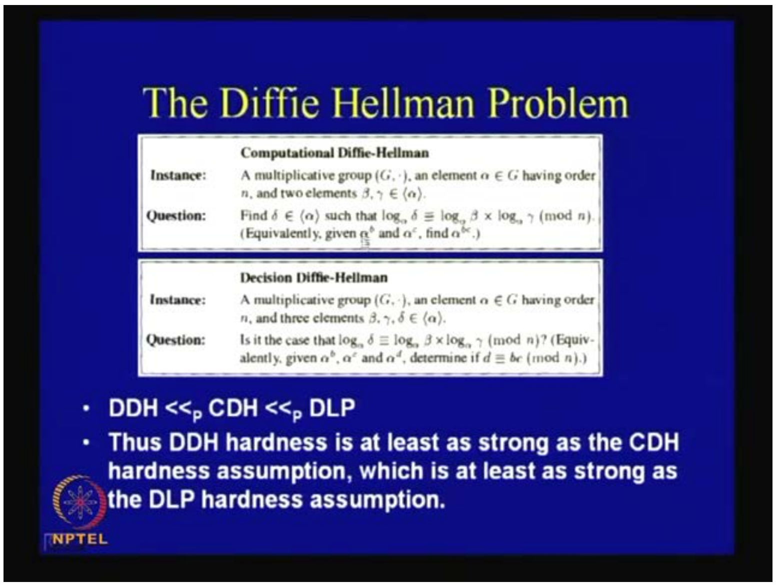
So, therefore, if from y 1 and y 2 the attacker is able to obtain the value of a; that means, it is able to solve the discrete logarithmic problem, then essentially, this cipher can be compromised and based upon the assumption that the discrete logarithmic problem is hard, we actually have got this cryptosystem which is popularly known as the ElGamal cryptosystem.

Now, in today's class we shall discuss about two things essentially. We shall discuss about that the discrete logarithmic problem is hard, but is it also difficult to obtain the bits of a discrete law. That is for example, if is it really difficult to obtain the 0 th bit, is it really difficult to obtain the first bit, is it really difficult to obtain i th bit, that is one part of the discussion.

The other part will be to reflect upon the cryptosystem of ElGamal with respect to symantec security. So, what is symantec security? So, we have seen that in context to some of the cipher's like adese is that, if we are given 2 plaintexts like x 1 and x 2 and we have given a ciphertext, so that means, we know that like either x 1 or x 2 has been encrypted, but we do not know which one has been encrypted.

Now, the problem is that from y that is a ciphertext, if we are able to kind of guess that whether x 1 has been encrypted or x two has been encrypted then, we say that it is a violation on the symantec security. So, therefore, a good cryptosystem, a cryptosystem which is semantically secured should also protect against such kind of information leakage. So, we shall study that whether the ElGamal cryptosystem also is belongs to such kind of semantically secured cipher.

(Refer Slide Time: 06:04)



## The Diffie Hellman Problem

**Computational Diffie-Hellman**

**Instance:** A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order $n$, and two elements $\beta, \gamma \in \langle \alpha \rangle$.

**Question:** Find $\delta \in \langle \alpha \rangle$ such that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$. (Equivalently, given $\alpha^b$ and $\alpha^c$, find $\alpha^{bc}$.)

**Decision Diffie-Hellman**

**Instance:** A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order $n$, and three elements $\beta, \gamma, \delta \in \langle \alpha \rangle$.

**Question:** Is it the case that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$? (Equivalently, given $\alpha^b$, $\alpha^c$ and $\alpha^d$, determine if $d \equiv bc \pmod{n}$.)

- DDH $<<_p$ CDH $<<_p$ DLP
- Thus DDH hardness is at least as strong as the CDH hardness assumption, which is at least as strong as the DLP hardness assumption.

NPTEL

So, we shall continue with this problem and also another we will discuss about another application on their of this discrete logged problems, it is to the something which is called as a diffie-hellman problem ok. Now, diffie-hellman problem is essentially a very interesting way of doing q exchanges often. So, first of all let us study what is the problem. So, in the problem generally comes in two flavors, one is the computational problem and other is a decisional problem.

So, therefore, it is important that that it is important therefore that, when you are discussing about the computational problem that essential to understand about what the problem is. So, the problem is in this case, we have been provided with a group which is G and their product here is dot defined as dot and we are provided with an element alpha which belongs to G and which has got an order of n and we are provided with two elements like beta and gamma which belongs to this group ok.

So, beta and gamma belongs belonging to this group, we know that beta is nothing alpha raise to some value and gamma is also alpha raise to some value right. Now, our question is that, the question is therefore, to find another value which is delta, which belongs to this group also such that, the logarithmic of delta to the base a is equal to or congruent to the logarithm of beta base alpha multiplied with the logarithm of gamma base alpha, we are all doing modular n computations.

So, the question is I mean it can equivalently which stated like this, that is we have been provide with suppose two values like alpha power of b and alpha power of c. So, therefore, assume that the beta and gamma as I have told you, that can also be written as alpha power of some value like b and similarly, gamma can also be written as alpha power of c right, it is alpha raise to some value ok.

Now, the question is given these two values like alpha power of b and alpha power of c can be compute alpha power of b c that is the problem. So, with the computational diffie-hellman problems says us that, given these group description and given these two elements like beta and gamma can be compute these value of log, can you find out the value of delta such that these equation will satisfied, ok.

The other one is you can equivalently, you stated like if we are provided with alpha power of b and alpha power of c can be compute the value of alpha power of b c. So, that is the computational diffie-hellman problem or as is often refers to as the C D H problem ok.

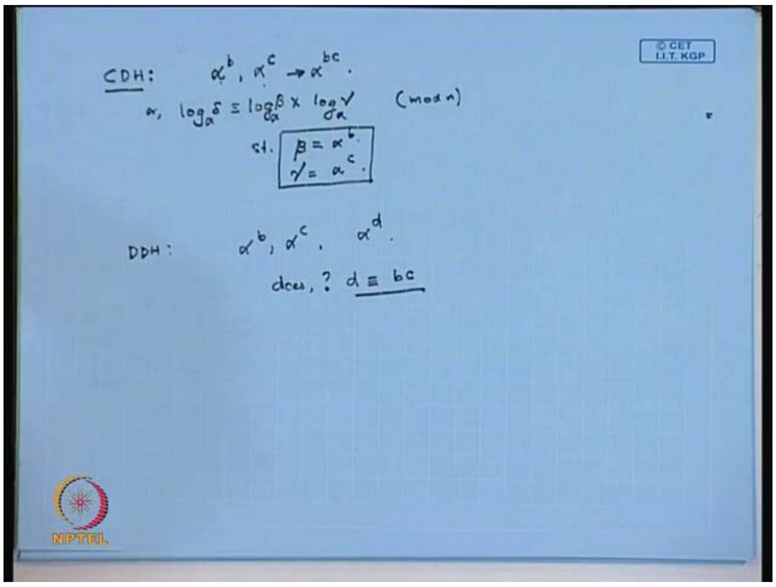
So, in this case we know that, we have we. So, we have. So, if I write it in a way is small way that is summarize, what is the computational diffie-hellman problem, it means that we are provided with two values like alpha power of b and we are provided with another value like alpha power of c can be compute the value of alpha power of b c or ok.

So, such that here beta essentially means alpha power of b and gamma means alpha power of c. So, therefore, here you have basically, you have supposed to compute the value of delta such that this equation is satisfied ok.

So, that means, here we are provided with two values like alpha power of b and alpha power of c can we compute the value of alpha power of b c, that is the computational diffie-hellman problem. Now, the decisional variant of this problem or as we call the decisional diffie-hellman problem or DDH is slightly different, the problem statement here is that, we are provided with alpha power of b and alpha power of c and we are also provided with another alpha power of d ok.

Now, the question is that, does d satisfy the value of b c. So, that is the problem that is does d satisfy the value of b multiplied by c. So, therefore, this is the question that is does d satisfy these particular relation.

(Refer Slide Time: 11:04)



CDH: $\alpha^b, \alpha^c \rightarrow \alpha^{bc}$.

$\alpha$, $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$

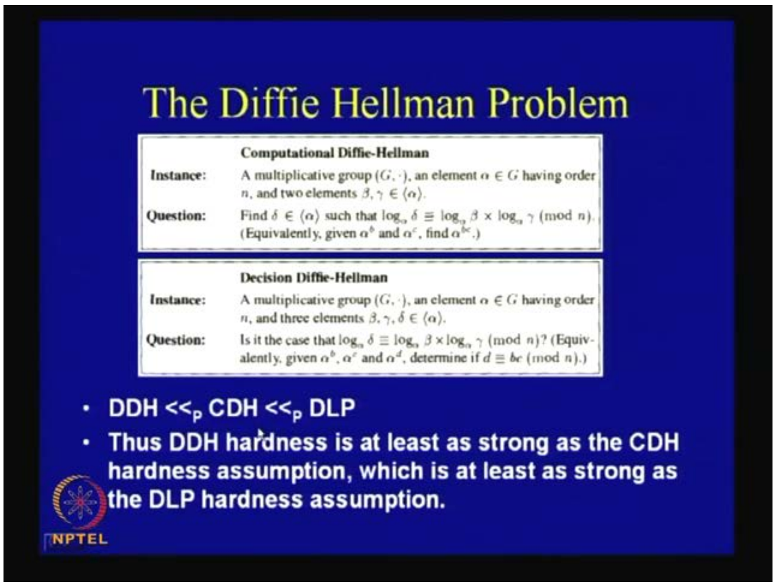
s.t. $\boxed{\begin{matrix}\beta = \alpha^b. \\ \gamma = \alpha^c.\end{matrix}}$

DDH: $\alpha^b, \alpha^c, \quad \alpha^d$.

does, ? $\underline{d \equiv bc}$

So, we can understand like I mean an obvious question would be like, so we have essentially seen 3 problems, right we have seen the discrete log problem, we have seen the computational diffie-hellman problem and we are seen also the decisional diffie-hellman problem. So, one obvious question is they are directive hardness, are all of them equivalent or as or is that one problem is more difficulty and than the other problem, right.

(Refer Slide Time: 11:34)



## The Diffie Hellman Problem

**Computational Diffie-Hellman**

|  |  |
|---|---|
| **Instance:** | A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order $n$, and two elements $\beta, \gamma \in \langle \alpha \rangle$. |
| **Question:** | Find $\delta \in \langle \alpha \rangle$ such that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$. (Equivalently, given $\alpha^b$ and $\alpha^c$, find $\alpha^{bc}$.) |

**Decision Diffie-Hellman**

|  |  |
|---|---|
| **Instance:** | A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order $n$, and three elements $\beta, \gamma, \delta \in \langle \alpha \rangle$. |
| **Question:** | Is it the case that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$? (Equivalently, given $\alpha^b$, $\alpha^c$ and $\alpha^d$, determine if $d \equiv bc \pmod{n}$.) |

- DDH $<<_p$ CDH $<<_p$ DLP
- Thus DDH hardness is at least as strong as the CDH hardness assumption, which is at least as strong as the DLP hardness assumption.

NPTEL

So, therefore, it can you can just see this statement which says that, the diffie-hellman problem. So, DDH, you can actually do polynomial, I mean in polynomial time or rather you can do reductions between these problems.

So, first of all, let us see that for example, if you have a solution to the discrete log problem then, the first of all try to deflect that you can actually solve the computational diffie-hellman problem, why? Because you see that, if you can solve the discrete log problem then, you can easily obtain can I mean for example, assume that if you can solve the computational diffie-hellman problem that is if the computational diffie-hellman problem is not hard.

So, then you can actually solve or rather if you assume that the diffie the discrete log problem is not hard, I mean I mean you can solve the discrete log problem, then you can actually solve the computational diffie-hellman problem.

Why, now because if we are provided, so what is the computational diffie-hellman problem challenge? We have been provided the alpha power of b and alpha power of c right. So, if you can solve the, if you have a mechanism to solve the discrete log problem, then from alpha power of b, you can compute the value of b. Similarly, from alpha power of c, you can compute the value of c. So, you can compute the value of b into c and therefore, you can also obtain the value of alpha power of b c right. So, this is trivial, you see that trivial if you can solve the discrete log problem, then you can solve the computational diffie-hellman problem.

So, that means, in terms of hardness we will see that the computational diffie-hellman problem, if it is hard then, that is if the assumption that the computational diffie-hellman problem is hard, then it also implies that the discrete log problem is hard ok.

So, therefore, we will say, that is one kind of one part of the understanding of the relative hardness. Now, you see the other part, that is if the computational diffie-hellman problem is not hard; that means if you have a mechanism to solve the computational diffie-hellman problem, then you also have a mechanism to solve the decisional diffie-hellman problem.

Now, these also very trivial, because you see that if you can... So, the this is your decisional diffie-hellman problem right that is you have been provided the alpha power of b, alpha power of c and alpha power of d and you have to check whether alpha power of d is equal to essential, I mean if d is congruent to b into c right. So, you see that, if you can solve the computational diffie-hellman problem, then given alpha power of b and alpha power of c, you can compute the value of alpha power of b c right.

So, now what you can do is that, you can just check that whether alpha power of b c is indeed equal to alpha power of d, right. If alpha power of b c is equal to alpha power of d then, you can conclude that d is indeed congruent to b into c, hence solve the decisional diffie-hellman problem.

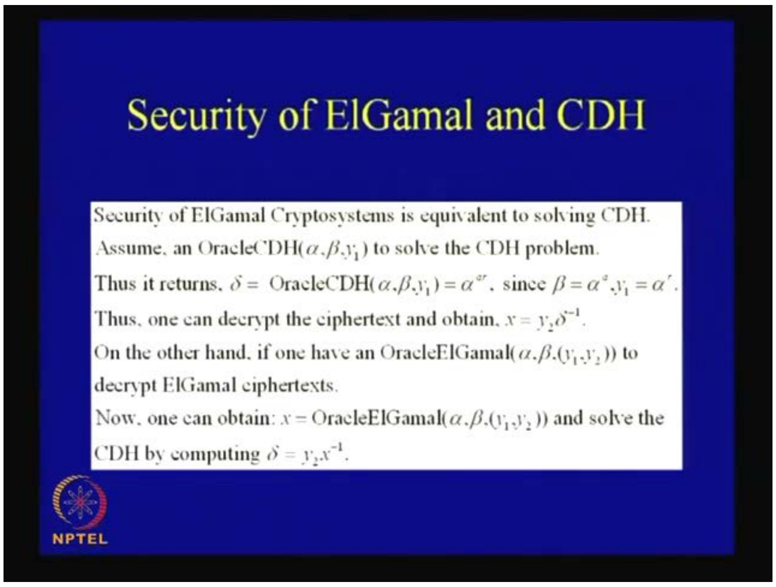
So, therefore, you can also similarly state that, if the assumption that the decisional diffie-hellman problem is difficult problem is a hard problem, then it implies that your computational diffie-hellman problem is also a difficult problem, right. So, therefore, we can actually conclude this line, that the DDH hardness is at least as strong as the computational diffie-hellman hardness assumption, which is at least as strong as the discrete logarithmic problem hardness assumption, right.

So, do you understand this? That is, so, therefore, what I am essentially doing is, then I am just writing this in kind of combining these two things and we are writing that the

DDH assumption implies that implies the CDH assumption, which implies the discrete log problem assumption ok.

So, that means, that these problem these assumption that is the DDH hardness is at least as strong an assumption as that of the computational diffie-hellman problem assumption, which is at least as strong an assumption as the discrete log problem assumption, right right ok.

(Refer Slide Time: 15:47)



## Security of ElGamal and CDH

Security of ElGamal Cryptosystems is equivalent to solving CDH.
Assume, an OracleCDH$(\alpha, \beta, y_1)$ to solve the CDH problem.
Thus it returns, $\delta =$ OracleCDH$(\alpha, \beta, y_1) = \alpha^{ar}$, since $\beta = \alpha^a, y_1 = \alpha^r$.
Thus, one can decrypt the ciphertext and obtain, $x = y_2 \delta^{-1}$.
On the other hand, if one have an OracleElGamal$(\alpha, \beta, (y_1, y_2))$ to decrypt ElGamal ciphertexts.
Now, one can obtain: $x =$ OracleElGamal$(\alpha, \beta, (y_1, y_2))$ and solve the CDH by computing $\delta = y_2 x^{-1}$.

So, now let us reflect another kind of interesting such equivalence or reduction, that the security of the ElGamal cryptosystems is equivalent to solving the computational diffie-hellman problem.

So, let us see so, therefore, if I say it is equivalent then, I have to show that if I can solve the ElGamal cryptosystem, then I can obtain a value of I can solve the computational diffie-hellman problem also. Similarly, the other way is that, if I can solve the computational diffie-hellman problem, I should have a mechanism to solve the ElGamal cryptosystem also.

So, here let us assume that, there is an oracle call CDH, which solves the computational dc element problem. So, what does it mean? That means if it is given, so that means, with I have an oracle, called oracle CDH, which takes a value of alpha beta and to y 1. So, what is y 1? Y 1 was equal to alpha power of r according to our ElGamal cryptosystem right. So, if I take a value of this thing, if I am I have been provide with this and what was beta equal to, beta was equal to alpha power of a right that is a secrete value.

So, now if I am able to solve the computational diffie-hellman problem, then this means they are from these two values, I should be able to compute the value of alpha power of a r, I should be able to compute the value of alpha power of a r. So, call this as delta; that means, you get a value of alpha power of a r using an oracle, which is able to solve the computational diffie-hellman problem.

So, now, one can actually decrypt and obtain the value of x using this, how? Now, because you know that, if you are able to obtain the value of alpha power of a r then, you need then alpha power of a r is nothing but, beta power of r right and you know that y 2 was equal to x multiplied by beta power of r right.

That is nothing but, x multiplied by delta right. So, therefore, in order to find the value of x, I just need to take y 2 and multiplied with the inverse of delta; that means, whatever the oracle to solve CDH problem gives me I take or compute the multiplied with the inverse and multiplied with y 2, I obtain the value of x, right.

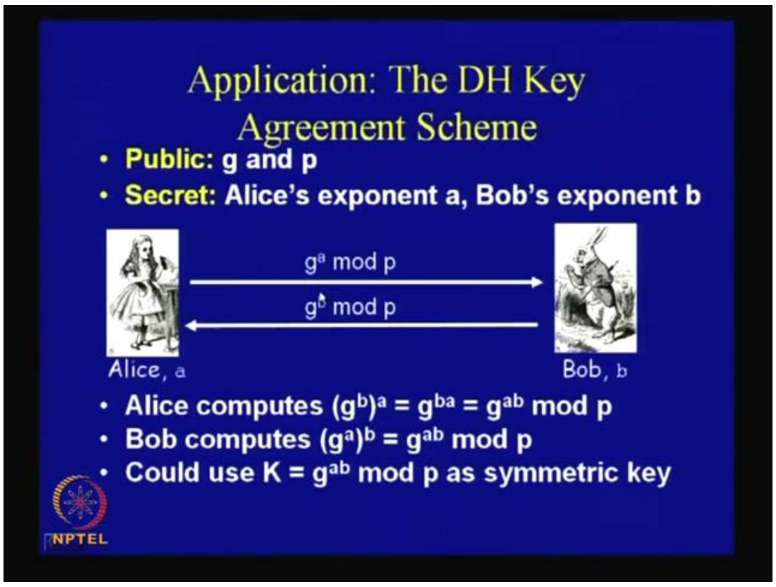
Now, that explains one direction of the problem, we have other direction to prove right. So, in that case what we will do? We will assume that, there is an oracle which solves the ElGamal cryptosystem right. So, assume that there is an oracle which solves the ElGamal cryptosystem and we take seen again two values like alpha beta and of course, the ciphertext y 1 and y 2 and gives back what, it gives back the value of x because that is the objective of this oracle, right.

If basically deciphers the value of the plaintext from y 1 and y 2. So, therefore, it returns back to me x. So, now, we know that, this relation is true, right we know that x is equal to y 2 into delta inverse. So, therefore, you see that trivially that, you can actually obtain the value of delta which is equal to y 2 into x inverse right and what is delta? We have discussed right that if you can calculate the value of delta, then you are essentially solving the computational diffie-hellman problem ok.

So, that means, that if you can solve the ElGamal cryptosystem, then you can also solve the computational diffie-hellman problem. So, therefore, you know that CDH assumption is equal to that ElGamal assumption and similarly, you see that the ElGamal assumption is also I mean is I mean implying so, therefore, the CDH assumption is implying the ElGamal assumption.

Similarly, the ElGamal assumption is implying the CDH assumption. So, therefore, we will see that the ElGamal assumption is equivalent to the CDH assumption, right it is clear.

(Refer Slide Time: 20:06)



## Application: The DH Key Agreement Scheme

- **Public:** g and p
- **Secret:** Alice's exponent a, Bob's exponent b

$g^a$ mod p
$g^b$ mod p
Alice, a
Bob, b

- Alice computes $(g^b)^a = g^{ba} = g^{ab}$ mod p
- Bob computes $(g^a)^b = g^{ab}$ mod p
- Could use $K = g^{ab}$ mod p as symmetric key

So, now, we shall go into an application of the diffie-hellman problem, it is a very interesting application of how to solve the key agreement scheme.

So, we have discussed like in context to symmetric ciphers, that you for example, if there is a network with say n nodes, and then you need to kind of exchange keys between all of them, right that is a huge amount of key exchange. So, which means that it is a very important problem to solve how to exchange the key is between two parties. So, we will we will see that, the discrete I mean the diffie-hellman problem essentially was propounded, I mean one of the strongest application of early of diffie-hellman problems is to solve the key agreement scheme that is to give a solution for key agreement.

So, we see that this is the setting like we have a public value of g and p, which are two public quantities and there is a secret, that is Alice has got, a secrete component called a and bob has a secrete component called b ok. So, therefore, what they do is this, there is alice computes g power of a modulo p, that is it takes the public quantities g, raises it to a and computes modulo p; the other the other part, the other is that bob also computes the value of g power of b modulo p and obtains and then a and obtains g power of b modulo p and sends it to alice ok.

So, now the objective of both alice and bob will be two way to reach a common consensus, that is to reach an agreed point. So, what alice does is this, that is alice takes g power of b modulo p because that is the part which if receives from bob and raises g power of b modulo p to its secrete value a ok.

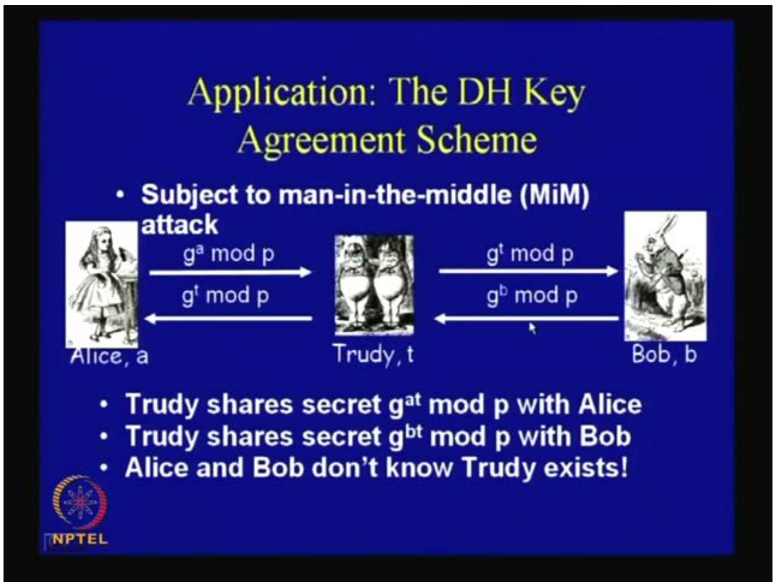
So, therefore, if I raise g power of b raise it to a, then I obtain g power of b a, but that is equal to g power of a b mode p right. So, what bob does is the other way? That is bob has got g power of a modulo p from alice and what it does is that, it raises it to the power of b. And therefore, uses of both alice and bob, essentially have computed in the value of g power of a b modulo p, right. Now, this is essentially used as a symmetric key for further encryption.

So, therefore, if I want to use and a s b s and other symmetric ciphers, then again use these exchanged key as my computed symmetric key right. So, now you see that, you can reflect upon the difficulty of that problem, that is why cannot a person who does not have a knowledge of this secrete values a and b, he is able to compute the value of g power of a b mod p ok.

So, you see that, that is precisely a diffie-hellman assumption, that is if we are provided with g power of a mod p and with g power of b modulo p, then I should not be able to complete the value of g power of a b modulo p, right or even I or the d c element the decisional d c element problems says as, there are even if I do not provided with a g power of b modulo p, I should not be able to say that, this g power b modulo p has that this nice property of being equal to g power of a b modulo. I should not be able to even decide that, that is a decisional diffie-hellman problem right.

So, we have a computational as I have flavor and you also have a decisional flavor of the same problem right. But, you see that, you can actually deflect that there is a very small I mean. So, these are a kind of something which is called as passive (( )), but what if it also can do an activate (( )). So, you see that the d c element key exchange actually does not give any god guard against that.

(Refer Slide Time: 23:51)



## Application: The DH Key Agreement Scheme

- Subject to man-in-the-middle (MiM) attack

$g^a$ mod p
$g^t$ mod p
$g^t$ mod p
$g^b$ mod p
Alice, a
Trudy, t
Bob, b
NPTEL

- Trudy shares secret $g^{at}$ mod p with Alice
- Trudy shares secret $g^{bt}$ mod p with Bob
- Alice and Bob don't know Trudy exists!

For example, you assume that there is party call trudy who comes in between and what it does is this. That is it takes g power of a modulo p and just change is it to something like g power of t modulo p and just change ok.

So, therefore, this g power of a modulo p is now not I mean I am not following the (( )) diffie-hellman problem, but I am just modifying it to something like g power of t modulo p. So, this kind of modifications is commonly referred in the cryptographic literature as man in the middle attacks. So, this is an active attack no doubt ok.

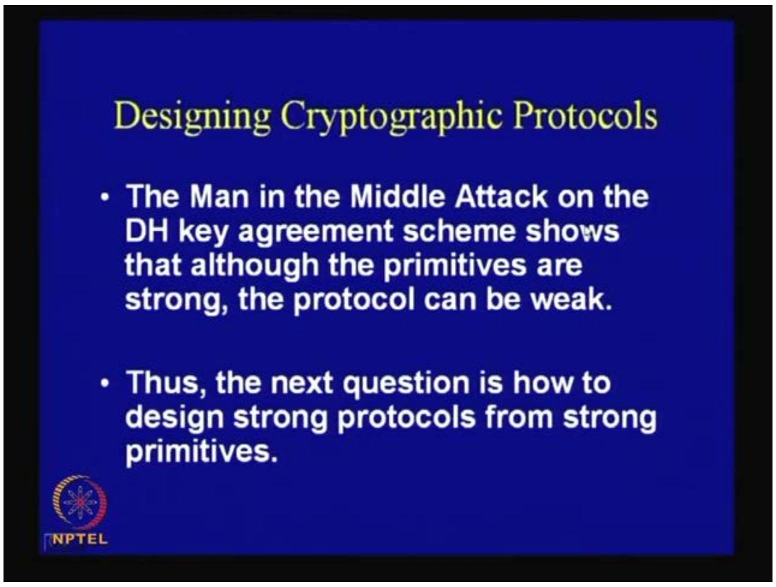
So, similarly, now you see (( )) g power of b modulo p, but trudy just modify something like g power of t modulo p, right. Now, you see that both of these parties what it will do is that, alice will raise this g power of t modulo p to its own secret. Similarly, trudy we can actually raise g power a modulo p to its own secret t right and obtain g power a t modulo p right. Now, what about trudy and bob, both of them actually computes the value of g power of b t modulo p.

So, now, you see that both alice and trudy can actually continue their communications using their secret key g power a t modulo p, what about trudy and bob? They can actually compute using I mean, continue communications using g power b t modulo p, but you note one thing that alice does not know the trudy exists, bob also does not know the trudy exists.

So, which means that it is a problem, right because you see that alice and bob are suppose to communicate between each other, while in the network actually alice is computing with a non trusted party or an attacker, without knowing that it is actually intruding its information. That is, knowing that the actual information or the actual intended receiver of the information is not receiving the information. So, this is a clear breach of the security right.

So, therefore, I see that this is the very something that we need to kind of guard against.

(Refer Slide Time: 25:58)

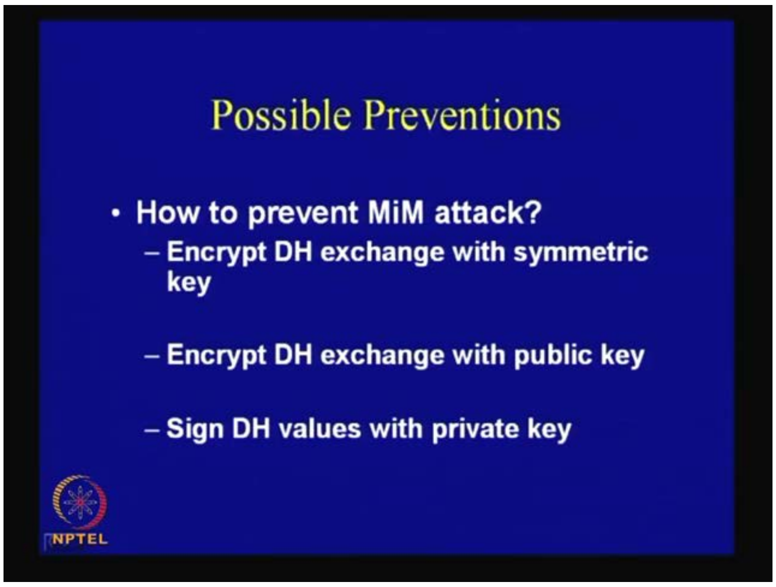


## Designing Cryptographic Protocols

- The Man in the Middle Attack on the DH key agreement scheme shows that although the primitives are strong, the protocol can be weak.
- Thus, the next question is how to design strong protocols from strong primitives.

So, therefore, the man in the middle attack on the diffie-hellman key agreement scheme shows that, although over primitives are strong your protocol can be weak right. So, we till now we have been actually trying to kind of design strong primitives right, but we see the here that, although your primitives are strong but, your application can be weak ok. So, therefore, the next question can come like, how to design strong protocols from strong primitives. So, that is a kind of a very kind of a pertinent research problem. So, here I mean, so, therefore, you have got the entire domain of cryptographic protocols and how to design protocols ok.

(Refer Slide Time: 26:36)

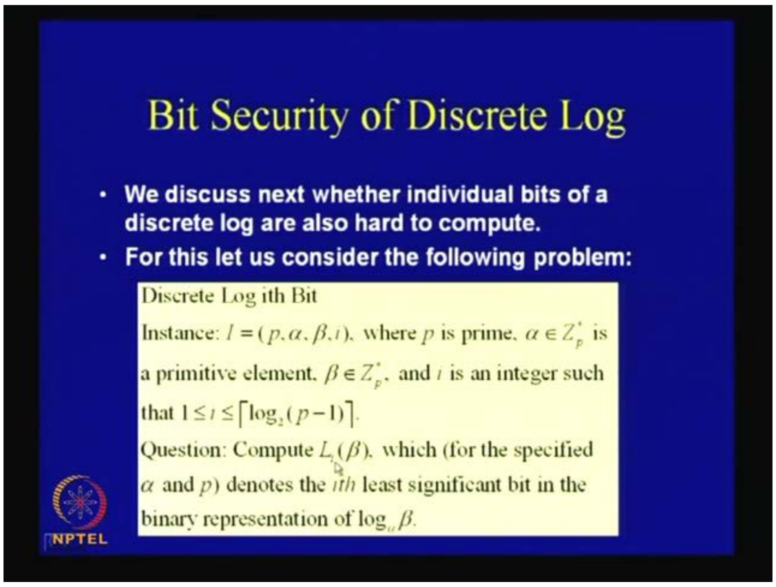


## Possible Preventions

- How to prevent MiM attack?
  - Encrypt DH exchange with symmetric key
  - Encrypt DH exchange with public key
  - Sign DH values with private key

So, there are some possible preventing techniques like for example, I will encrypt the diffie-hellman exchange with symmetric key, I will encrypt the diffie-hellman exchange with public key, I will sign the diffie-hellman problems with private keys. So, there are can be some after thoughts like how to prevent the man in the middle attack, but the point is that, remember that the diffie-hellman key exchange as stated as for the basic definition is not secured against man in the middle attack, you have to do something extra ok.

(Refer Slide Time: 27:06)



## Bit Security of Discrete Log

- We discuss next whether individual bits of a discrete log are also hard to compute.
- For this let us consider the following problem:

Discrete Log ith Bit

Instance: $I = (p, \alpha, \beta, i)$, where $p$ is prime, $\alpha \in Z_p^*$ is a primitive element, $\beta \in Z_p^*$, and $i$ is an integer such that $1 \le i \le \lceil \log_2(p-1) \rceil$.

Question: Compute $L_i(\beta)$, which (for the specified $\alpha$ and $p$) denotes the $ith$ least significant bit in the binary representation of $\log_\alpha \beta$.

But anyway, we will not continue with this right now, but we rather continue with the bit discrete log problem and reflect upon the mathematical problem of the bit security of discrete logs ok.

So, what we will first discuss is this problem, that is the discrete log i th bit problem. So, what is the problem first of all? The problem is that we have been given instance like again what we have seen previously prime number p, alpha, beta and the value of i. Here, your alpha belongs to z p star and is a primitive element, beta also belongs to z p star and i is an integer, which lies between 1 and logarithm of p minus 1 base 2.

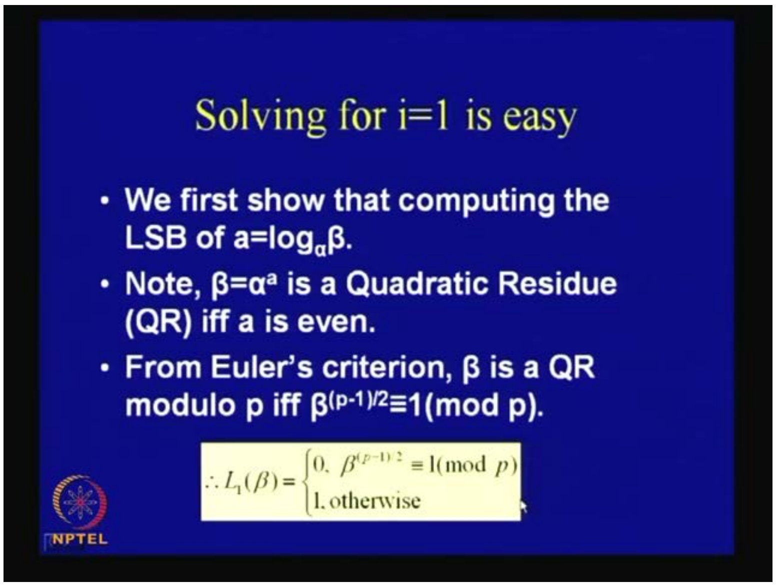
The problem that we are considering right now or that we are referring as the discrete log I th bit problem is this, that is to compute l I beta that is given beta and the beta is equal to alpha power of a, the problem is that to essentially compute the value of the i th least significant bit in a binary representation of log beta base alpha ok.

So, your logarithm of beta base alpha is denoted as a right. Now, you know that a can be represented using a binary values. So, I can start like this x 0, x 1 and so on, right. So, my problem when I am considering the l I beta is to compute the value of x I or rather x I minus 1 ok.

So, my problem is to essentially compute the value of the i th p. So, when I am referring to l 1 beta, I am compute the value of x 0, when I am referring to l I beta I am computing the value of x I minus 1, so that is the problem. So, you can say that, this x 0 I mean that depends upon your numbering; you can also number it from x1 to x2 and so on, just to avoid confusion. So, you can say that, l 1 beta, actually refers to x1 in that case and x I minus 1 refers to x I, that is just to avoid the confusion of this the indices ok.

So, anyway the problem is to compute the i th least significant bit.

(Refer Slide Time: 29:30)



## Solving for i=1 is easy

- We first show that computing the LSB of $a=\log_\alpha\beta$.
- Note, $\beta=\alpha^a$ is a Quadratic Residue (QR) iff a is even.
- From Euler's criterion, $\beta$ is a QR modulo p iff $\beta^{(p-1)/2}\equiv 1 \pmod p$.

$$\therefore L_1(\beta)=\begin{cases}0, & \beta^{(p-1)/2}\equiv 1 \pmod p\\ 1, & \text{otherwise}\end{cases}$$

So, we can actually see that, solving for i equal to 1 is very easy, that is to compute the computing the least significant bit of a, that is where a is equal to log beta base alpha is actually very easy, so, why? So, for this we will actually use our previous idea of something which is called as quadratic residues.

So, we have studied quadratic residues in contest 2 RSA right, we discussed about RSA. So, what was quadratic residue? Quadratic residue any one was said to be quadratic residue, if you can actually represent it by the square of another element right in that group. So, then we refer that to as a quadratic residue. So, therefore, if I say that beta equal to alpha power of a, then if this is a quadratic residue, then this is actually if and only if condition is that, a has to be even ok.

So, therefore, you need to compute, so, therefore, beta will be a quadratic residue if and only if, a is even. And from Euler's criteria, which we have studied also previously, we know that if beta I mean beta is a quadratic residue of modulo p if and only if, the way of checking that is by raising beta to the power of p minus 1 by 2 and checking that whether it is congruent to one modulo p right. So, what can be the possible values of beta to the power p minus 1 by 2, it can be either plus 1 or it can be minus 1, because you see that from farmar's little principle beta power of p minus 1 is equal to 1, right.

So, it can be either plus 1 or minus 1. So, you see that if it is equal to plus 1 modulo p then, beta is actually a quadratic residue; similarly, beta is a quadratic residue implies that, this is an if and only if condition necessary and sufficient right. So, you see that therefore, if I plot these 2 points, what does it mean? It means that, I have to obtain l 1 beta right.

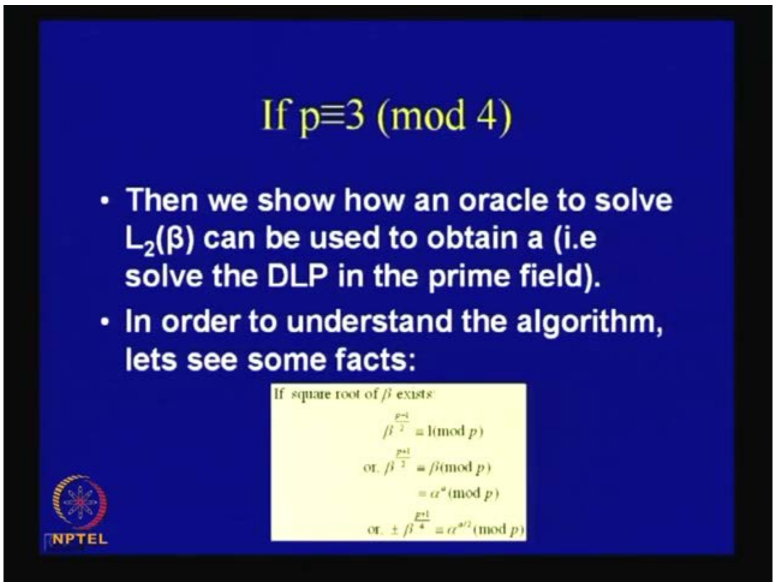
So, we have to computing the least significant bit. Now, if this is an even value, that is if a is even then, what is the value of l 1 beta? It is equal to 0, because only then the exponent is even right. So, you see that if it is even, right what it says is that, then this is an if and only if condition, it is equivalent to saying that beta is a quadratic residue and if beta is a quadratic residue, then its equivalent to compute these value.

So, therefore, what will you do is that you will take beta and raise it to the power of p minus 1 by 2 and check whether it is equal to 1 modulo p congruent to 1 modulo p, if that is so, then you conclude that a is even and therefore, l 1 beta is 0, right. Otherwise it is odd and hence one right you see this is an efficient way of understanding the 0 th p or the first p or have been the least significant p.

So, that means, what? That means, the although a discrete log problem is believed to be a difficult problem, but actually the 0 th bit is easy to compute or the least significant bit is easy to compute. So, you see that these two problems are difficult, where are that different like why that the that computing the value of the discrete log problem is not exactly the same as compute of an I th bit of the discrete log right. So, you see (( )) it is very easy to compute the least significant.

Now, the question is it also difficult to compute any bit the first bit, second bit and so on because you see that there is a danger right because if I am able to retrieve all the bits then essentially I can solve the discrete log problem right. So, therefore, the question is whether the next bit is also difficult or the next bit is also difficult ok.

(Refer Slide Time: 33:08)



## If p≡3 (mod 4)

- Then we show how an oracle to solve $L_2(\beta)$ can be used to obtain a (i.e solve the DLP in the prime field).
- In order to understand the algorithm, lets see some facts:

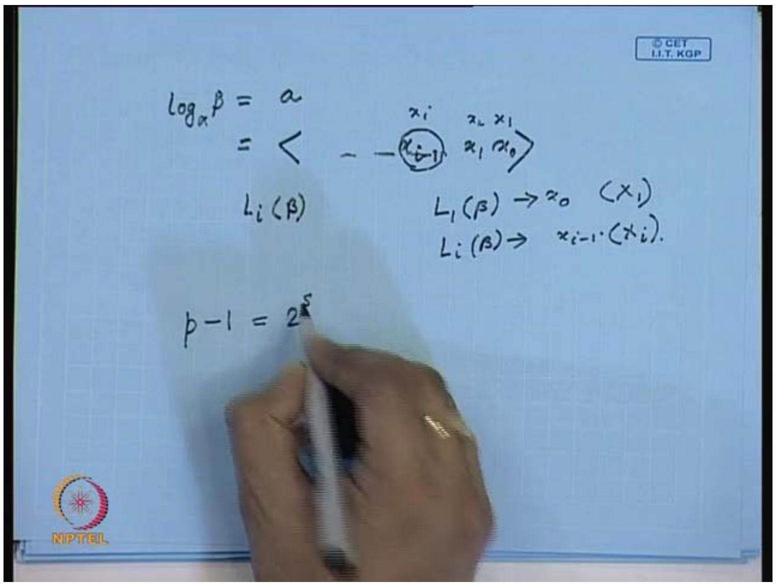
If square root of $\beta$ exists:

$$\begin{aligned} \beta^{\frac{p-1}{2}} &\equiv 1 \pmod p \\ \text{or, } \beta^{\frac{p+1}{2}} &\equiv \beta \pmod p \\ &\equiv \alpha^{a} \pmod p \\ \text{or, } \pm\beta^{\frac{p+1}{4}} &\equiv \alpha^{a/2} \pmod p \end{aligned}$$

So, we will first of all consider a special case, where p is congruent to 3 modulo 4. So, in this case why why we are talking about p being congruent to 3 modulo 4, I will talk about that later, but first of all let us discuss about this case like p is congruent to 3 modulo 4.

So, you note that if you are if p is congruent to 3 modulo 4 then I mean, so, for example, if you for example, if you can refer like p right you can always right like p minus 1 for example, as p minus 1 is a even value right.

(Refer Slide Time: 33:44)



$\log_\alpha \beta = a$

$= \langle \,-\, - \boxed{x_{i-1}}\; x_1\, x_0 \rangle$, with $x_i$ above $x_{i-1}$ and $x_2\; x_1$ above $x_1\; x_0$

$L_i(\beta)$

$L_1(\beta) \rightarrow x_0 \quad (x_1)$

$L_i(\beta) \rightarrow x_{i-1} \cdot (x_i).$

$p-1 = 2^s$

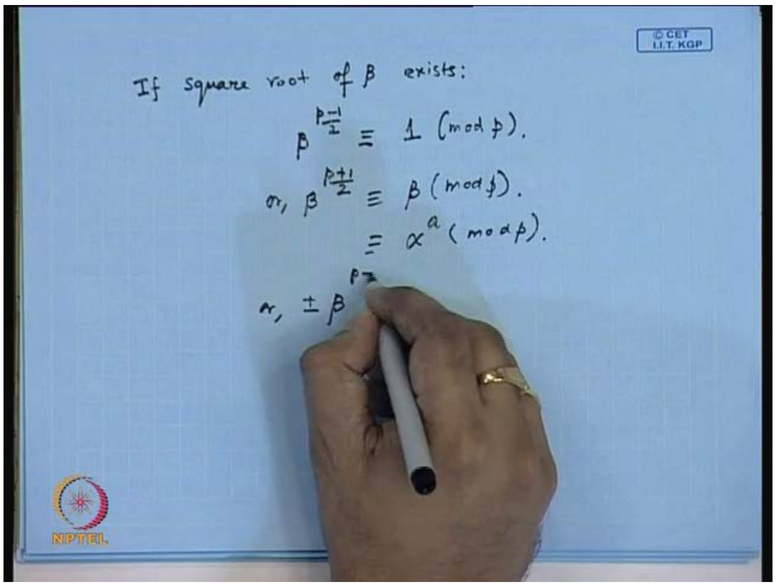
So, therefore, you can always write it as some 2 power of s right multiplied with some t value where t is not a multiple of 2 that is t is an odd value right . So, therefore, if I take the base, I mean the minimum case where s, so s is greater than equal to 1. So, I take the s case s equal to 1 if s is equal to 1, then it means that p minus 1 is equal to 2 into t, right.

So, you see that this p equal to that means, that p is congruent to 3 modulo 4 because if you take p and if you subtract like 1. So, it becomes 2, right. So, then it becomes 2 modulo 4. So, therefore, this is the best I mean the first case, when we consider this case like p minus 1 equal to 2 power of s t, then I am considering the case that s is equal to 1 and hence I am considering a case when p is congruent to 3 modulo 4 ok.

So, therefore, if p is congruent to 3 modulo 4 then, we will see a very interesting case; that is, if we have an oracle to solve the value of l 2 beta, that is compute the value of l 2 we have seen that l 1 beta is easy right. So, if we actually can we compute l 2 beta then, we can use that to obtain the value of the discrete, that is we can solve the discrete log problem in the prime field ok.

So, now in order to understand the algorithm, we have to see some facts we have to prove some facts. So, first of all let us try, let us start with this that is, let us start with this that is, if the square root of beta exists, that is beta is a value which is given to me and believe that, assume that the square root of beta exists.

(Refer Slide Time: 35:30)



If square root of $\beta$ exists:

$$\beta^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

$$\text{or, } \beta^{\frac{p+1}{2}} \equiv \beta \pmod{p}.$$

$$\equiv \alpha^{a} \pmod{p}.$$

$$\text{or, } \pm\beta^{p}$$

So, if the square root of beta exists, then we know that beta is a quadratic residue right; that means beta power of p minus 1 by 2 is congruent to 1 modulo p from the Euler's criteria, right. So, what we do is that, we multiply now both sides by beta, if you multiply both sides by beta, then you have got on the left side beta p plus 1 by 2 because you multiply by beta. So, if plus 1 gets added and on the right hand side, you have got beta modulo p right.

Now, what is beta modulo p? We know that beta modulo p is nothing but, alpha power of a modulo p right and you know that if and only if that I mean beta is a quadratic residue if and only if, a is even right. So, which means a by 2 exists and is an integer. So, therefore, we can say that, plus minus beta p plus 1 by 4 is congruent to alpha power of a by 2 modulo p right; that means, if the square root of beta exists then, the square root can be obtained by computing plus minus beta to the power of p plus 1 by 4.

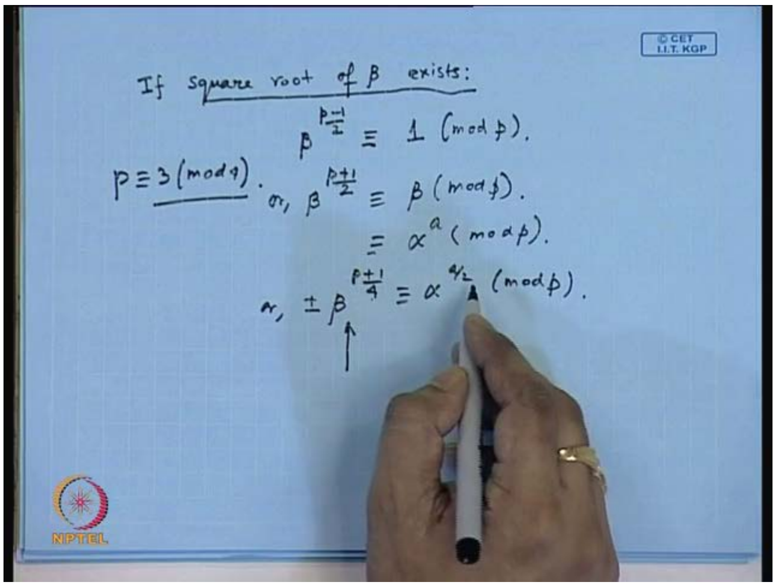
So, now note that we are talking about the case when p is equal congruent to 3 modulo 4 right. So, therefore, you see that p plus 1 by 4 is actually an integer, right and that is why we actually, if I want to compute the square root of beta, then I can actually compute beta to the power of p plus 1 by 4, plus or minus, is it ok. So, you see that if p was not equal to 3 modulo 4, then I would not have been share that it is an integer and the hence I could not have say that this the square root ok.

But here, since this value is given to us, we can say that, if I want to compute the square root, I can compute beta to the power of p plus 1 by 4 plus or minus or the square roots right. And we also wrote one thing, that is if I compute the value of some value like take any value like gamma square and p minus gamma square is same, when you are doing a modulo p and p is a prime right. So, that means, you have got essentially exactly p minus 1 by 2 quadratic residues we discussed about this, right and these 2 values are different like this gamma and this p minus gamma are different ok.

So, therefore, if you have got this plus so, therefore, if I am if I want to compute the square root, then I know that, it is either plus beta to the power of p plus 1 by 4 or it is minus beta to the power of p plus 1 by 4 right. And we also know that, if one of them is gamma, then the other one will be p minus gamma because there are only 2 squares, right if the square roots exist, then there are 2 squares ok.

So, therefore, the question is that I know that, alpha power of a by 2 is congruent to plus minus beta to the power of p plus 1 by 4 modulo p, but the question is which one is correct? That is which is the actual square root, so for that we will use this factor that is you see that I know that, l 2 beta is the second bit of beta right.

(Refer Slide Time: 37:07)



If square root of $\beta$ exists:

$$\beta^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

$p \equiv 3 \pmod{4}$.

$$\text{or, } \beta^{\frac{p+1}{2}} \equiv \beta \pmod{p}.$$
$$\equiv \alpha^{a} \pmod{p}.$$
$$\text{or, } \pm\beta^{\frac{p+1}{4}} \equiv \alpha^{a/2} \pmod{p}.$$

© CET I.I.T. KGP
NPTEL

Now, when you are computing alpha power of a by 2 right, then essentially it is a first bit. So, can you see that l 1 alpha to the power of a by 2 is actually actually equal to l 2 to the power of I mean l 2 beta, right because you know that, when you are taking the square root, right then essentially, so first of all you know that, the square root exists. That means, if you think in terms of like thinking a bit like x 0, x 1 and so on, and then your x 0 is 0 because the square root exists, right. So, for example, if I write this a, a could be what? a would be like something like x 0 plus 2 x 1 like so on, right.

So, therefore, if I know that, since the square root of alpha power of a exists; that means, this is equivalent to say in that is the square root of this exists, equivalent to saying that a is even, so that means, is equivalent to saying that x 0 is 0. That means, your a is actually this plus 2 x 1; that means, you can compute the value of a by 2 and that will be equal to this plus x 1, right.

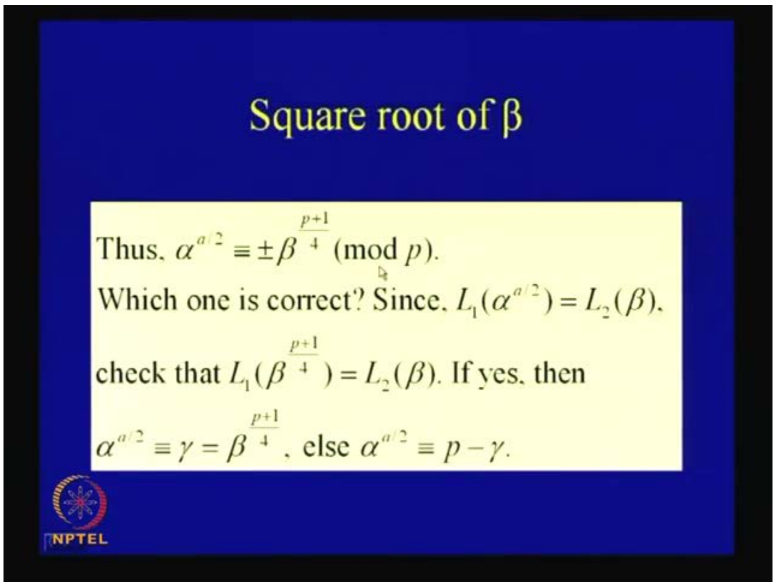
So, therefore, what is the value of l 1 of alpha power of a by 2 that will be the value of this or this, right that is the first bit, that is equal to x 1. And this is exactly equal to that of l 2 of beta because in that case, this one was a second bit right. So, therefore, l 1 alpha power of I mean, l 1 alpha power of a by 2 is equal to l 2 beta right.

So, therefore, what you can do is that, if you can check that if I just check this, so, I know that beta to the power p plus 1 by 4 is a probable square root, I compute l 1 of that and check whether it is equal to l 2 beta, right if I can compute l 2 beta of course, right. So, if yes then alpha power of a by 2 will say is congruent to gamma, which is equal to beta to the power of p plus 1 by 4 else, half a power of a by 2 will be congruent to p minus gamma because that is a other probable square root right, so that means, either this is the square root or this is the square root, agreed on this.

So now, we will actually see the steps to obtain the entire discrete block. Now, you know that a which is equal to log beta base alpha is nothing but, this is equal to x i and you multiply it with two power of i right to obtain the entire decimal value of a. So, therefore, you see, so you see that alpha power of… So, if you obtain the value of beta then, beta is equal to alpha power of like something like x 0 plus 2 x 1 plus so on, right.

So, therefore, l 1 of beta, we can actually use l 1 of beta because we saw that it was an easy problem to obtain the value of x 0. So, one can easily obtain the value of x 0 by the previous technique that we have seen, right.

(Refer Slide Time: 38:53)



## Square root of β

Thus, $\alpha^{a/2} \equiv \pm\beta^{\frac{p+1}{4}} \pmod{p}$.

Which one is correct? Since, $L_1(\alpha^{a/2}) = L_2(\beta)$,

check that $L_1(\beta^{\frac{p+1}{4}}) = L_2(\beta)$. If yes, then

$\alpha^{a/2} \equiv \gamma = \beta^{\frac{p+1}{4}}$, else $\alpha^{a/2} \equiv p - \gamma$.

So, now the question is how to I mean now the question is that, we have discussed that we will assume that, somebody can compute the second beta, can compute the value of l 1 beta either can compute the value of l 2 beta and from there try to obtain the value of a. So, therefore, what we will do is this, that is we will take, we will adopt a method like this; we will take beta and compute the value of beta divided by alpha power of x 0.

So, if I take beta and divide beta by alpha power of x 0, then that is equal to what? Alpha to the power of 1 plus 2 x 1 modulo p, so basically, what we have done is that, just taken out the part for alpha power of x 0, this is the part which remains, right.

So, now the square root of beta exists for sure, why because you know that this is an even value, the exponent is even right. So, what we do is that, we compute the value of gamma which is equal to beta to the power of p plus 1 by 4. So, note that beta has been updated by these values, do you see that? So, we compute the value of beta to the power of p plus 1 by 4.

Now, you obtain I mean, in order to obtain the actual square root because you know that, there is a dilemma because this is plus beta power of beta power of p plus 1 by 4 or p minus that. So, what we do is that, we compute x 1 from the oracle to compute l 2 beta. So, assume that there is an oracle which actually computes the value of l 2 beta. So, that is the way how we do the proofs, right.

So, we assume that there is an oracle which computes the value of l 2 beta and gives the value of x 1. So, now, we can check that whether x 1 is indeed equal to l 1 gamma, right we are actually obtain the value of gamma and I know that, computing l 1 gamma is easy. So, I can actually use compute the value of l 1 gamma and check whether it is equal to the value of oracle l 2 beta returns right.

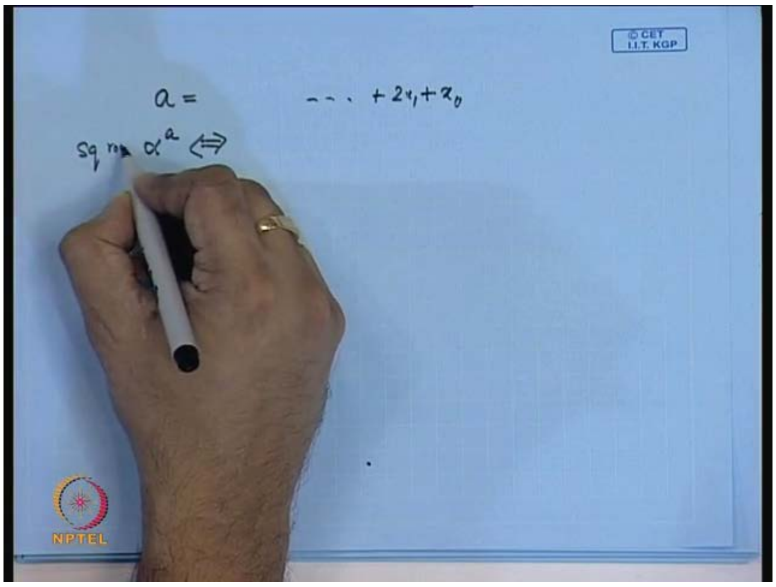
If this is so, then the square root of beta is indeed beta to the power of p plus 1 by 4, else it is equal to p minus that right. So, basically at each stage, you are sure about what is this square root, right. So, therefore, you can continue this method like this. So, you see that we continue this process of the oracle beta to obtain the bits x 2, x 3 and so on to obtain the value of the discrete log in this fashion, ok.

So, summarizing this is the algorithm, the final algorithm that is what you do is this. That is you take so, therefore, the objective of this is using l 2 oracle to solve the discrete log problem. So, I have been given with p alpha and beta, my objective is to compute the value of a. So, what I first do is that, I obtain the value of x 0 by computing the value of l 1 beta right.

So, then I compute the value of beta is equal to beta by alpha power of x 0 modulo p, because I know already a value of x 0 I can compute alpha power of x 0 and then I start a running index I equal to 1 and until and unless while beta is not equal to 1 what we do is this. We calculate the value of x I which is equal to what oracle l 2 beta returns and compute the value of beta to the power of p plus 1 by 4, ok.

And now, we check that, where that if l i gamma is equal to x i if l i gamma is equal to x i, we will say that beta is equal to gamma because then beta is a square root, otherwise p minus gamma is a square root .Then we again compute the value of beta by dividing beta by alpha power of x i so that means, you see that at each stage, we are actually taking out we are computing the square root, right.

(Refer Slide Time: 39:58)



$a = \quad \ldots + 2x_1 + x_0$
sq rt $\alpha^a \Leftrightarrow$

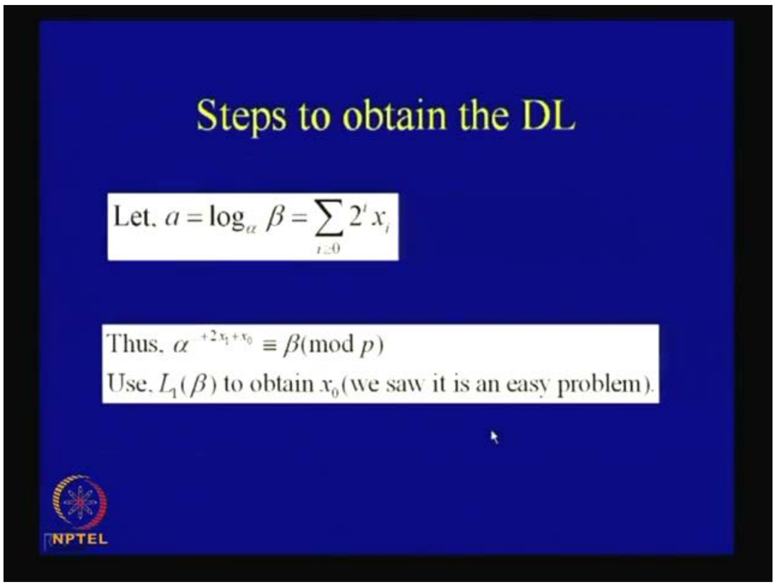
So, that means that, if you have to been given like say for example, alpha power of so on. So, for example, first of all we have got like plus 2 x 1 plus x 0, you have divided this by alpha power of x 0 that was the first step, right. And from there you compute the value of alpha x 1 plus 2 x 1 and you know that, the square root existed because this one was an even quantity, right.

And then, what did you do? You basically took out I mean you confirm that, you confirm the square root of this, right. and then what and What will be the square root of this? The square root of this essentially will be like you can you essentially have computing the you are raising this value to the power of half, right and once you compute the square root of this, again you have got like something like alpha this thing plus x 1, right.

So, you have got 2 here, you have got 2 x 2, you see that. So, now, if I divide again this by alpha power of x 1 I again have this right, alpha so on plus 2 of x 1 right. So, that means, sorry 2 of x 2, two of x 2 right, so that means, I again can I again can I know that again that the square root of this exist, because this is again an even number, right.

So, that means, again I can compute confirm the value of square root and I know that, if I confirm the value of square root, then I am again raising this to the power of half right. So, again it becomes something plus x 2. So, again I divide that alpha power of x 2 right. So, what you are basically doing is that, using this oracle which solves the l 2 beta, you are actually confirming the value of the square root, ok.

(Refer Slide Time: 41:53)



## Steps to obtain the DL

Let. $a = \log_{\alpha} \beta = \sum_{i \geq 0} 2^i x_i$

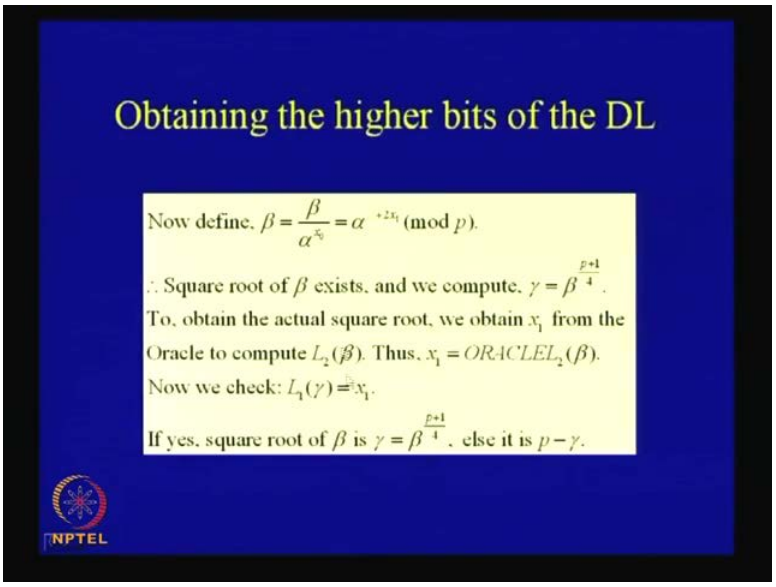
Thus. $\alpha^{\dots + 2x_1 + x_0} \equiv \beta (\bmod p)$

Use. $L_1(\beta)$ to obtain $x_0$ (we saw it is an easy problem).

And from there, you are actually dividing that contribution and computing the discrete logs, gradually one after the other you are obtaining the bits and you are if you are able to obtain all the bits, then you essentially solve the discrete log value.

So, that is a basic algorithm to solve the discrete log problem. Now, some final comments on this would be like, the prove holds for any p, where p is equal to for p minus 1 is 2 power of s 3. So, we does we told you that this is actually hold for any p minus 1, where p minus 1 is equal to 2 power s into t. So, it does not necessarily hold only for p is congruent to 3 modulo 4, we can actually have a similar result when p minus 1 is congruent or rather is equal to 2 power of s t, where s is an integer ok.

(Refer Slide Time: 43:01)



## Obtaining the higher bits of the DL

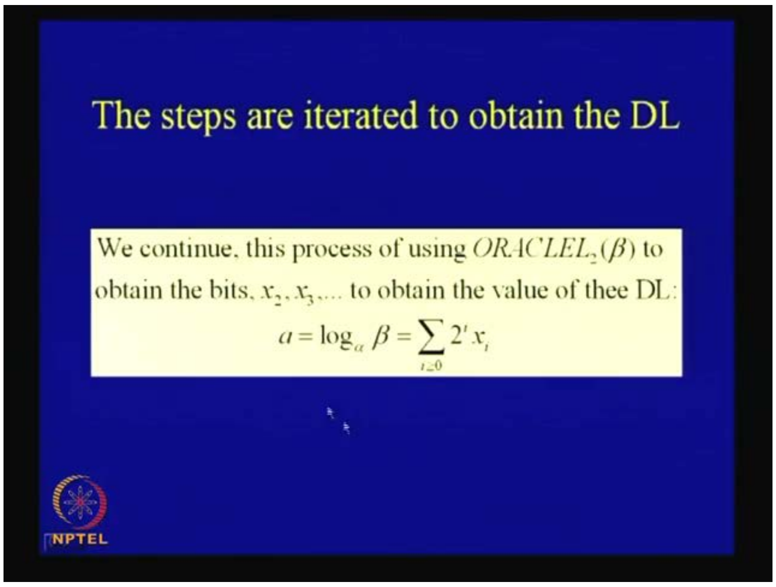
Now define, $\beta = \frac{\beta}{\alpha^{x_0}} = \alpha^{+2x_1} \pmod{p}$.

$\therefore$ Square root of $\beta$ exists, and we compute, $\gamma = \beta^{\frac{p+1}{4}}$.
To, obtain the actual square root, we obtain $x_1$ from the Oracle to compute $L_2(\beta)$. Thus, $x_1 = ORACLEL_2(\beta)$.
Now we check: $L_1(\gamma) = x_1$.

If yes, square root of $\beta$ is $\gamma = \beta^{\frac{p+1}{4}}$, else it is $p - \gamma$.

So, in that case, we will see that we can I mean using the previous thing we have reflect that, obtaining s l s b's will be easy, because you now that, p minus 1 is equal to 2 power s t right. So, therefore, using the similar technique, you can obtain s l s b's x 0, x 1 x 2 till x's minus 1. But, however, obtaining the x plus 1 th bit will be difficult, I mean if you are able to obtain the x plus 1 th bit, right then exactly similarly what we have seen in context to p equal to congruent to 3 modulo 4, you can obtain the entire discrete log ok.

(Refer Slide Time: 44:54)



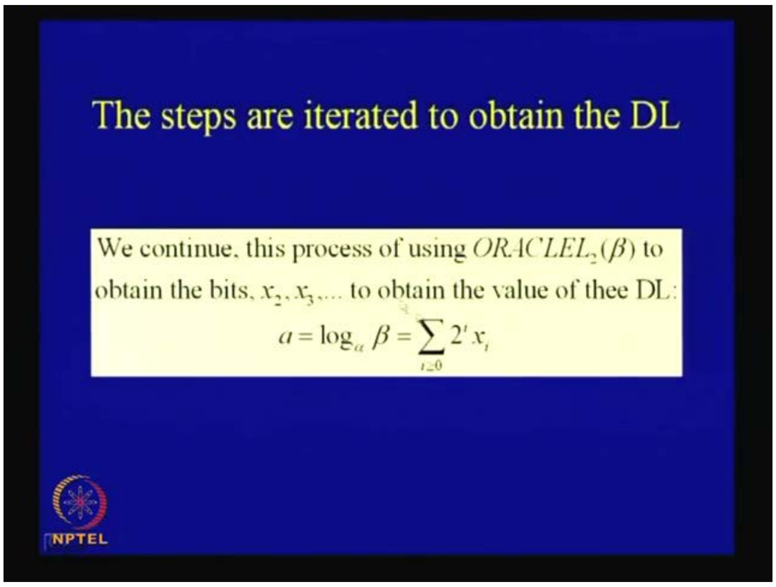
## The steps are iterated to obtain the DL

We continue, this process of using $ORACLEL_2(\beta)$ to obtain the bits, $x_2, x_3, \ldots$ to obtain the value of thee DL:

$$a = \log_\alpha \beta = \sum_{i=0} 2^i x_i$$

So, therefore, if you I mean the discrete log problem here, it will be difficult for this kind of beta, when i is going from 0 to s, I mean to s minus 1, but So, therefore, here in this part it will be easy, this will be easy, but l I beta will be hard, when i is greater than equal to s, it will be difficult. So, it is difficult, again the hardness is difficult in the sense that if you are able to solve this problem, then you are able to solve the discrete log problem, ok.

(Refer Slide Time: 44:54)



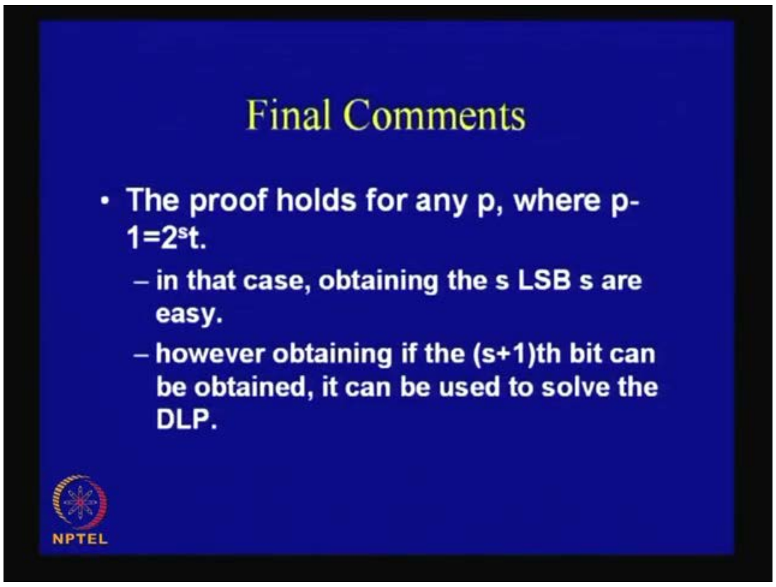
## The steps are iterated to obtain the DL

We continue, this process of using $ORACLEL_2(\beta)$ to obtain the bits, $x_2, x_3, \ldots$ to obtain the value of thee DL:

$$a = \log_\alpha \beta = \sum_{i=0} 2^i x_i$$

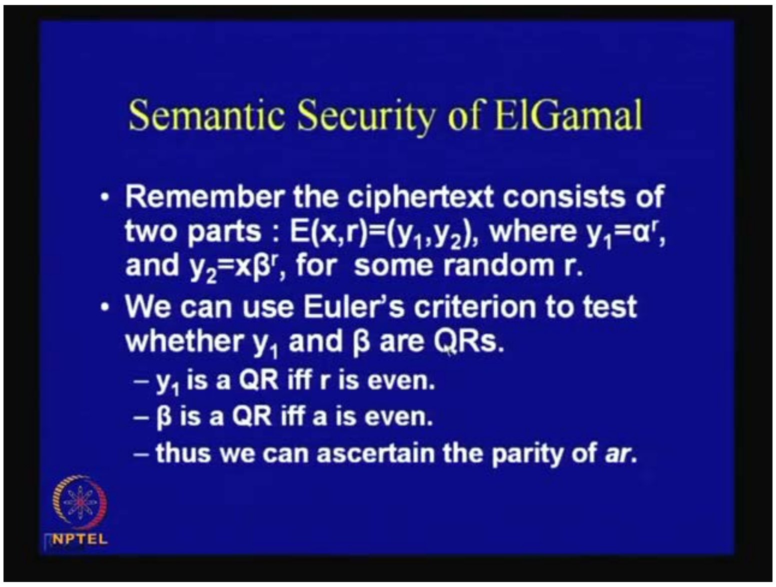
So, when I say hard it is an assumption, it is an assumed hardness, right is this part clear.

(Refer Slide Time: 50:30)



## Final Comments

- The proof holds for any p, where p-1=$2^s$t.
  - in that case, obtaining the s LSB s are easy.
  - however obtaining if the (s+1)th bit can be obtained, it can be used to solve the DLP.

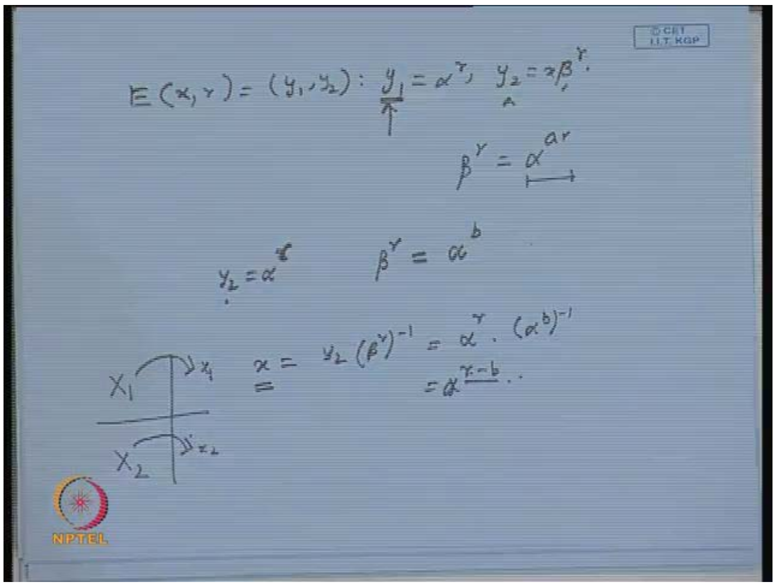


## Semantic Security of ElGamal

- Remember the ciphertext consists of two parts : $E(x,r)=(y_1,y_2)$, where $y_1=\alpha^r$, and $y_2=x\beta^r$, for some random r.
- We can use Euler's criterion to test whether $y_1$ and $\beta$ are QRs.
  - $y_1$ is a QR iff r is even.
  - $\beta$ is a QR iff a is even.
  - thus we can ascertain the parity of *ar*.

So, now, we shall conclude by some comments under semantic security of ElGamal crypto systems. So, in this case, now objective is that, I have been provided with the ciphertext for two different plaintexts and I have to distinguish, right at the say from the ciphertext are whether x 1 has been encrypted or whether x 2 has been encrypted. So, for that, let us just little bit recap the ElGamal crypto system.

(Refer Slide Time: 51:03)



$$E(x, r) = (y_1, y_2) : \underset{\uparrow}{y_1} = \alpha^r, \; y_2 = x\beta^r.$$

$$\beta^r = \alpha^{ar}$$

$$y_2 = \alpha^r \qquad \beta^r = \alpha^b$$

$$x = y_2 (\beta^r)^{-1} = \alpha^r . (\alpha^b)^{-1} = \alpha^{\underline{r-b}}$$

$X_1$
$x_1$
$X_2$
$x_2$
© CET I.I.T. KGP
NPTEL

We know that, E x comma r is equal to y 1 comma y 2, where y 1 is equal to alpha power of r and y 2 is equal to x multiplied with beta power of r, for some random r, ok.

So, now we can use the Euler's criteria to test what, to test that to test y whether y 1 and beta are quadratic residues, how? So, what you can do is this right, you can have actually you know that I mean as a as an attacker, you have an access to y 1 right. So, therefore, what will you do is that, you will raise y 1 to the power of p minus 1 by 2 and check that whether it is equal to 1 or not and from there, you can conclude that whether y 1 is a quadratic residue or not.

Now, if y 1 is so you know that, y 1 is a quadratic residue is equivalent to saying that r is even number. So, you can derive the parity of r, right. Similarly, you know that, beta is equal to alpha power of a, I mean it is a public variable. So, you can again compute beta power of b minus 1 by 2 and from there you can conclude that whether beta is a quadratic residue or not. Now, if you know that, beta is a quadratic residue again from there you can obtain the information that, whether a is even or not right.

So, therefore, using Euler's criteria from y 1 and from beta you can obtain the parity of both r and a right. And therefore, you can obtain the parity of a r right. So, you can obtain the parity of a r means, then you can actually obtain you can obtain some more information like, you know that if a r is an even value, then you know that, beta power of r is nothing but, alpha power of a r. So, from there, you can also conclude that whether beta power of r is a quadratic residue element right.

So, therefore, one can actually engage use this Euler's criteria to understand that, whether beta power of r is a quadratic residue or not, do you see this. So, if beta power of r is a quadratic residue, then I can always write beta power of r as some value of alpha power of b, where b is even, right.

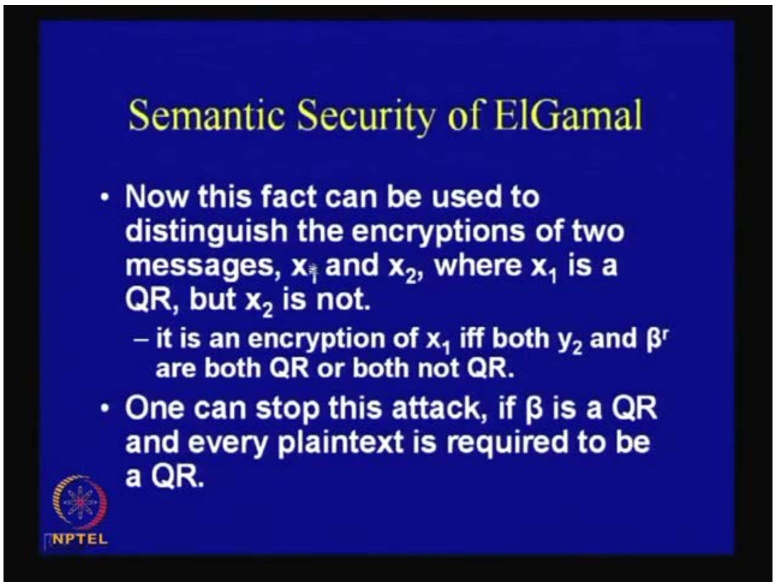
Now, the other thing is that you know that, if so, similarly you also know that, y 21 is some alpha power of you can say alpha power of c or all that you know it is alpha power of r, so you also know that from the fact that, whether y 2 has is an has it is a quadratic residue or not, you know that also that whether r is even or not right and what is x? x is nothing but, y 2 multiplied with the inverse of beta right.

So, what is y 2? Now, y 2 is alpha power of r multiplied with the inverse of alpha power of b right. So, that is equal to alpha power of r minus b, yes or no right. So, now, you see that if x is a quadratic residue, then you know that r minus b has to be even right.

So, which means that either both r and b are odd or both r and b are even, right; so which means that, if I have got two classes of plaintext. Say one class of plaintexts, where all of them are quadratic residues and the other class, where both of them are non quadratic residues. So, suppose I choose one value of x 1 from here and I choose 1 value of x 2 from this class of x 2, right. I will do an encryption, ElGamal encryption on x 1 and I will do an ElGamal encryption of x 2.

Now, from this I can actually obtain understand that, whether x 1 has been encrypted or whether x 2 has been encrypted, because what I will check is the parity of these things and try to see that, whether both r and b are even or both r and b are odd. If both an r and b are even and both r and b are odd, then it indicates, so if either both r and b are even and or both r and b are odd, that it indicates that x is a quadratic residues, do you see that; otherwise, x is a non quadratic residue.

(Refer Slide Time: 55:34)



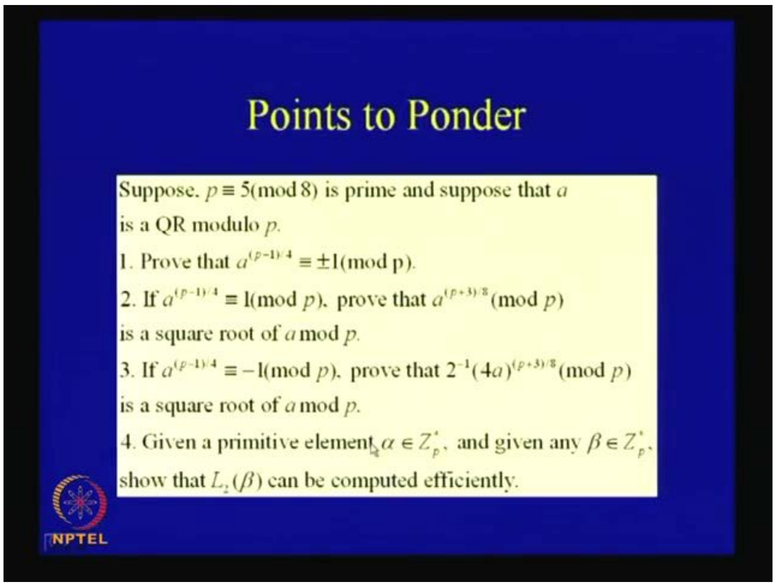
## Semantic Security of ElGamal

- Now this fact can be used to distinguish the encryptions of two messages, $x_1$ and $x_2$, where $x_1$ is a QR, but $x_2$ is not.
  - it is an encryption of $x_1$ iff both $y_2$ and $\beta^r$ are both QR or both not QR.
- One can stop this attack, if β is a QR and every plaintext is required to be a QR.

So, summarizing this is what it says here that now, this fact can be used to distinguish the encryptions of two messages x 1 and x 2, where x 1 is a quadratic residue, but x 2 is not. Now, it is an encryption of x 1, if and only if both y 2 and beta power of r are both quadratic residues and both non quadratic residues.

So, one can stop this attack, if beta is a quadratic residue at every plaintext is required to be a quadratic residues. So, you see that the some extra condition needs to be imposed to make ElGamal encryption also semantically secure, right.

(Refer Slide Time: 56:07)



## Points to Ponder

Suppose, $p \equiv 5 \pmod 8$ is prime and suppose that $a$ is a QR modulo $p$.

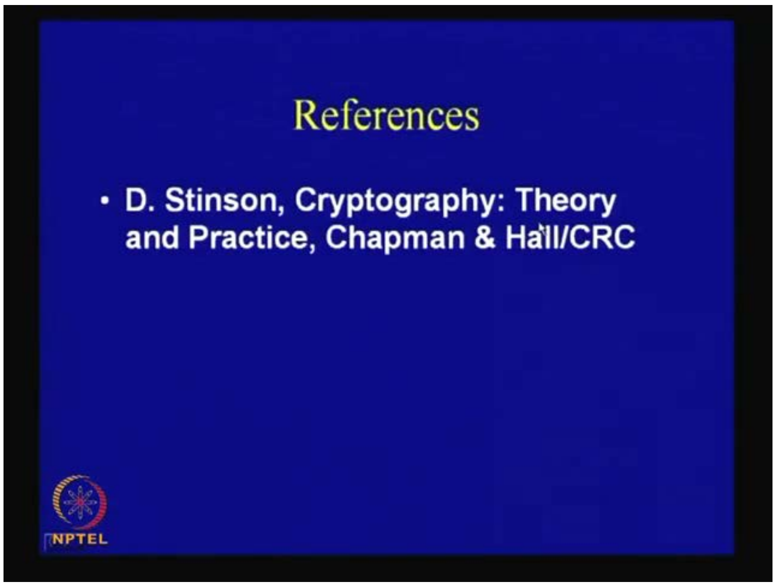
1. Prove that $a^{(p-1)/4} \equiv \pm 1 (\text{mod p})$.
2. If $a^{(p-1)/4} \equiv 1 \pmod p$, prove that $a^{(p+3)/8} \pmod p$ is a square root of $a$ mod $p$.
3. If $a^{(p-1)/4} \equiv -1 \pmod p$, prove that $2^{-1}(4a)^{(p+3)/8} \pmod p$ is a square root of $a$ mod $p$.
4. Given a primitive element $\alpha \in Z_p^*$, and given any $\beta \in Z_p^*$, show that $L_2(\beta)$ can be computed efficiently.

So, I will give you one point to kind of some exercise to solve, like we have talked about beta is congruent to 3 modulo 4. So, you can also take this problem of beta being congruent to 5 modulo 8.

And suppose that a is congruent residue modulo p, suppose one problem what you can do is that you can show that, a power of p minus 1 by 4 is congruent to plus minus 1 modulo p and if a power of p minus 1 by 4 is congruent to 1 modulo p then, you can prove similar to what we have done actually. That is prove that a power of p plus 3 by 8 modulo p is a square root of a mod p; otherwise; if it is minus 1, then you can show that 2 power of minus 1 into 4 power 4 a raise to the power of p plus 3 by 8 modulo p is a square root of a mod p.

And given a primitive element alpha, which belongs to z p star and given any beta which belongs to z p star, you can show that l 2 beta can be actually computed efficiently. So, this is an exercise, which you can take and which you can try to solve.

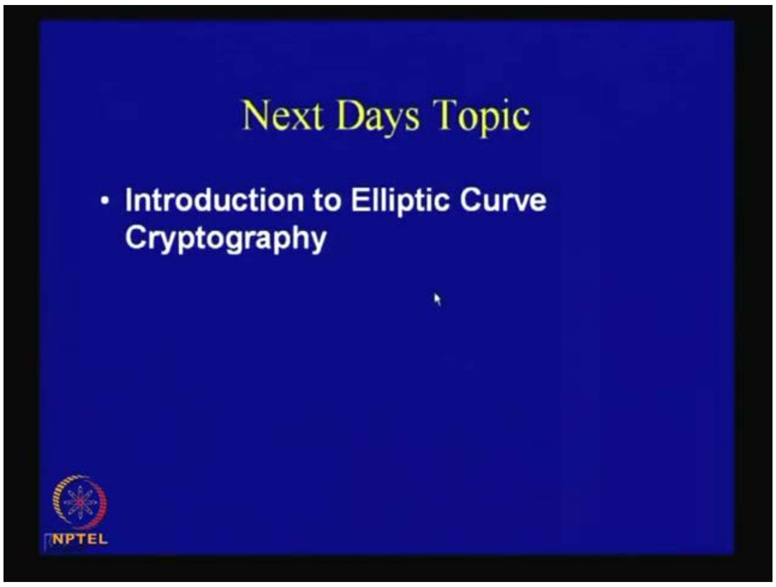
(Refer Slide Time: 57:14)



## References

- D. Stinson, Cryptography: Theory and Practice, Chapman & Hall/CRC

(Refer Slide Time: 57:23)



## Next Days Topic

- Introduction to Elliptic Curve Cryptography

So, my references are been from Stinson's book that is of for Stinson cryptography theory and practice of Chapman and hall and C R C, you can get more details of what we have discussed. And in next day's topic, we will introduce the topic of elliptic curve cryptography.