

**Secure Computation: Part 1**  
**Prof. Ashish Choudhury**  
**Department of Computer Science & Engineering**  
**International Institute of Information Technology-Bengaluru**

**Lecture - 40**  
**Discrete Logarithm and DDH Assumption**

Hello everyone, welcome to this lecture.

**(Refer Slide Time: 00:34)**

---

### Lecture Overview

- Cyclic groups
  - ❖ Discrete logarithm problem
  - ❖ Decisional Diffie-Hellman problem
    - Variant of the Decisional Diffie-Hellman problem

---

So we had seen how to construct bit OTs where the sender's inputs were bits using RSA assumption and in general based on any one way trapdoor permutation and its corresponding hardcore predicate or hardcore function. We will now shift our attention or to design oblivious transfer protocols where the sender's input could be strings, they are no longer single bits.

So for that, we will first recap from our foundations of crypto course, the concepts related to Diffie-Hellman problem, what exactly is a cyclic group, Diffie-Hellman problem, discrete log problem, discrete log assumption, Diffie-Hellman assumption and so on. So this will be a quick lecture because this will be a quick recap of the concepts.

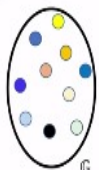
If you want to know more about these concepts you are referred to the NPTEL course on foundations of cryptography.

**(Refer Slide Time: 01:31)**

## Cyclic Groups $\{g, g^2, \dots, g^{q-1}\} = G$

□ **Definition** :  $(G, \circ)$  is a **cyclic group of order  $q$**  if the following hold:

- ❖  $(G, \circ)$  satisfies the **group axioms** --- closure, associativity, identity element, existence of the inverse element for each group element
- ❖ The set  $G$  consists of  $q$  elements ---  $|G| = q$
- ❖ **Existence of a generator** --- there exists at least one special element  $g \in G$ , such that all elements of  $G$  can be **generated** by different "powers" of  $g$



□ Let  $p$  be a prime and  $\mathbb{Z}_p^* \cong \{1, \dots, p-1\}$

- ❖ **Multiplication modulo  $p$**  --- for every  $a, b \in \mathbb{Z}_p^*$ :  $(a \cdot_p b) \cong [ab] \pmod p$
- **Theorem (Number Theory)**: For every prime  $p$ ,  $(\mathbb{Z}_p^*, \cdot_p)$  is a cyclic group of order  $p-1$
- **Theorem (Number Theory)**: Let  $(G, \circ)$  be a group of order  $q$ , where  $q$  is a prime. Then
  - ❖  $(G, \circ)$  is a cyclic group
  - ❖ All the elements of  $G$ , except the identity element is a generator

So let us start with cyclic group, what is a cyclic group? We say that set  $G$  along with the abstract operation  $\circ$  is a cyclic group of order  $q$  if the following conditions hold. First of all, the set  $G$  along with the operation  $\circ$  has to be a group. It has to satisfy the group axioms namely the closure property, the associativity property, the existence of the identity element, existence of the inverse element for each group element so on.

Then since we are saying that the order of the group is  $q$ , by that we mean that the cardinality of the set  $G$  is  $q$ , there are  $q$  number of elements. And by cyclic we mean that there exists at least one special element little  $g$  in the set big  $G$ , which we call as a generator. And it is generator in the sense that by raising or by computing different powers of that element little  $g$  we can generate all the elements of your set big  $G$ , right?

That means, if I take if I compute  $g$  to the power 0,  $g$  to the power 1,  $g$  to the power 2 and all the way to  $g$  to the power  $q-1$ , then these elements are nothing but the various elements of my set big  $G$ , okay. So there are several examples, nice examples of cyclic groups and specifically cyclic groups which we encounter in the cryptographic applications.

So if  $p$  is a prime, then the set  $\mathbb{Z}_p^*$  is denoted by the set, it is a collection of the elements 1 to  $p-1$ . And we are already aware of this multiplication modulo  $p$  operation. So if you want to perform a multiplication  $b$  modulo  $p$  then we first multiply  $a$  and  $b$  and then take mod with respect to modulo  $p$ .

Now we can use a well-known fact from number theory which says that this collection  $Z_p$  along with the operation multiplication modulo  $p$  is a cyclic group if  $p$  is a prime number. And its order is  $p - 1$  because there are  $p - 1$  elements in this collection. In general we can prove that if you are given a group of prime order okay, suppose the group order is  $q$  there are  $q$  number of elements in the set big  $G$ .

And if the number  $q$  is a prime number then there are some nice properties for this group. We can prove that this group is always a cyclic group and more importantly you do not have to separately look for the generators. You take any element from the set big  $G$  except the identity element. It is bound to be a generator for this group big  $G$ , okay.

(Refer Slide Time: 04:34)

### Discrete Logarithm in Cyclic Groups

□  $(G, \circ)$  be a cyclic group of order  $q$  --- without loss of generality, let it be multiplicative

❖ Let  $g$  be a generator for  $G$

$\{g^0, g^1, \dots, g^{q-1}\} = G$

❖ Let  $y$  be any arbitrary element of  $G$

➤ There exists a unique  $x \in \{0, 1, \dots, q - 1\}$ :

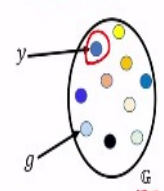
$g^x = y$

➤ The unique  $x \in \{0, 1, \dots, q - 1\}$  is called the discrete logarithm of  $y$  with respect to  $g$  --- denoted as  $DLog_g y = x$

❖ Discrete logarithms obey the rules of natural logarithms

$DLog_g e = 0$      $DLog_g (h^r) = (r DLog_g h) \pmod q$      $DLog_g (h_1 h_2) = (DLog_g h_1 + DLog_g h_2) \pmod q$

❖ Theorem (Number Theory): If  $g^x = y$  for some arbitrary integer  $x$ , then  $DLog_g y = [x \pmod q]$



$a^x = y$   
 $\Rightarrow \log_a y = x$

So now next let us quickly go through the concept of discrete logarithm in cyclic groups. So what is discrete logarithm? So again assume you are given a cyclic group of order  $q$  and this is an abstract operation small  $\circ$  which could be interpreted in the additive sense or it could be interpreted in the multiplicative sense without loss of generality. I will use the multiplicative interpretation of this abstract group operation.

Now since there are  $q$  number of elements in this set big  $G$  if I take, and it is a cyclic group, this means that by computing different powers of this generator, I can obtain all the elements of the set big  $G$ , that is a property of the generator because the

element is small  $g$  is a generator of your set big  $G$ . Now that means that if I take any element  $y$ , any element from the set big  $G$  okay.

Since small  $g$  is a generator, that means definitely there exist some power  $x$  in the range  $0$  to  $q - 1$  such that the generator power  $x$  will produce the element  $y$ . Why so? Because I know that a small  $g$  is a generator. So raising small  $g$  to different powers in the range  $0$  to  $q - 1$ , I can produce all the elements of the set big  $G$ . And since small  $y$  is an element of big  $G$ , definitely one of these powers is a small  $y$ .

So call that power as  $x$ , right? So that power  $x$  in the range  $0$  to  $q - 1$  such that the generator power that  $x$  gives you the element  $y$  is called as the discrete logarithm of the element  $y$  with respect to the generator  $g$ , okay. And we use this notation to denote the discrete logarithm. So this is very much similar to the concept of natural logs in we know that if  $a$  to the power  $x$  is equal to  $y$ , then we say that  $\log$  of  $y$  to the base  $a$  is equal to  $x$ .

We are just extending that concept in the context of cyclic groups, because the base will be now my generator, because the generator raised to different powers can generate all the elements. You take any element from the set big  $G$ , the discrete logarithm is the unique power  $x$  in the set  $0$  to  $q - 1$  such that generator power that  $x$  gives you the element  $y$ .

And interestingly like your natural logarithms, discrete logarithms also obey similar rules. So for instance, if I take the identity element is small  $e$ , so this is small  $e$  is the identity element of the group, abstract group, big  $G$ . So if I take the discrete logarithm of identity element to the base generator, we get  $0$  because  $g$  to the power  $0$  is defined to be the identity element in the context of cyclic groups, okay.

On the other hand, if I take an element  $h$  from the set big  $G$  raised to the power  $r$  and then try to compute the discrete logarithm of the resultant element, it will be same as  $r$  times the discrete logarithm of  $h$  modulo  $q$ . Why modulo  $q$ ? Because  $r$  times the discrete logarithm of  $h$  might cross  $q$ . But the discrete logarithm always are in the range  $0$  to  $q - 1$ .

So that is why if I take modulo  $q$ , I get the discrete logarithm of the element  $h$  to the power  $r$ . Whereas if you are given two group elements  $h_1$  and  $h_2$  and if you are multiplying them and then trying to find out the discrete logarithm, then it is equivalent to computing the discrete logarithm of the two elements, adding them and then taking modulo  $q$ .

More importantly, if you are given that if generator power  $x$  is equal to  $y$  and if  $x$  is not in the range  $0$  to  $q - 1$ , suppose it is more than  $q - 1$ , then you can obtain the discrete logarithm of  $y$  by computing  $x$  modulo  $q$ . So all this results I am taking from my earlier course on foundations of cryptography. I am assuming you are aware of these results. If you want to know more about these results, you are referred to the foundations of cryptography course.

(Refer Slide Time: 09:21)


### Discrete Logarithm Problem and Assumption

□ DLog problem ---- to efficiently compute the DLog of a random group element

Publicly known group description of a cyclic group ---  
 $(\mathbb{G}, o, \hat{g}, g): |\mathbb{G}| = \lambda$ , with  $\text{poly}(\lambda)$ -time algorithms for performing group operations (and exponentiations)


Experiment:  $\text{DLog}_{\mathbb{G}, \hat{g}}(\lambda)$

$\alpha \in_r \{0, \dots, q-1\}$



$u \leftarrow g^\alpha$

$\alpha' \in \{0, 1, \dots, q-1\}$



PPT  $\mathcal{A}$

$\text{DLog}_{\mathbb{G}, \hat{g}}(\lambda) \cong 1$ , if and only if  $g^{\alpha'} = u$

□ **Definition (DLog assumption):** Dlog assumption holds in  $(\mathbb{G}, o)$ , if for every PPT  $\mathcal{A}$  there is a function  $\text{negl}(\lambda)$ :  
 $\Pr[\text{DLog}_{\mathbb{G}, \hat{g}}(\lambda) = 1] \leq \text{negl}(\lambda)$

□ Several candidate groups, where DLog assumption is strongly believed (but not yet proved) to be true

So now we have introduced a discrete logarithm definition. Now let us see how easy or how difficult it is to compute the discrete logarithm of a random element. So the discrete logarithm problem or the DLog problem is the following. You are given the description of the group. You are given a generator. And you are given a random element from the group. The challenge is to efficiently compute the discrete logarithm of that random element.

And by efficiently I mean in polynomial amount of time. Polynomial in the number of bits that we need to represent a group elements. So it turns out that there are certain candidate cyclic groups for which solving the DLog problem for random instances is

believed to be difficult because we do not have any efficient algorithm. So to formalize that we introduced the DLog experiment with respect to a security parameter  $\lambda$ , okay.

This is security game placed between a verifier, hypothetical verifier and an hypothetical polynomial time adversary or algorithm who would like to solve the DLog problem for random instances. The public information which is available in this game are the group description and by group description I mean, the description of the group operation, its order, its generator.

And we assume that each element of the group here requires  $\lambda$  bits to be represented, okay. Now a challenge instance is created for this adversary by picking a random element from the group. How it can be done? So if you want to pick a random element from the group big  $G$ , then randomly pick an index in the range  $0$  to  $q - 1$ . Why  $0$  to  $q - 1$ ?

Because the order of the group is  $q$ , that means starting from generator power  $0$  to generator power  $q - 1$  you can obtain all the elements of the group. So to pick a random element of the group just pick a index randomly from the set  $0$  to  $q - 1$ . Keep the index with yourself and give the adversary the element  $u$  which is  $g$  to the power  $\alpha$ . And computing  $g$  to the power  $\alpha$  is efficient, because it can be done using the square and multiply approach.

Now the challenge for this adversary is to find out the  $\alpha$  or the discrete log of the element  $u$  in polynomial amount of time. So after analyzing the element  $u$  it has to submit a response. We do not know what is the strategy this adversary or the algorithm is following for computing the discrete logarithm. Whatever is the strategy it outputs an index in the range  $0$  to  $q - 1$ .

And we say that the adversary has won the experiment or the output of the experiment is  $1$  if and only if and indeed  $\alpha$  prime is the discrete logarithm of  $u$ , namely  $g$  to the power  $\alpha$  prime, gives the element  $u$ . And the discrete log assumption definition is the following.

We will say that the discrete log assumption holds if indeed for any polynomial time algorithm  $a$ , who participates in this experiment, the probability that he can solve the discrete log instance or when the experiment is upper bounded by some negligible function in the security parameter. And as I said earlier, there are several candidate groups for which we believe that the DLog assumption is indeed true.

That means there exists no polynomial time algorithm for solving a random instance of discrete logarithm in those groups. But as I said, strongly believe it is not mathematically proved. That is why it is only conjecture that the DLog problem is difficult to solve in those groups.

(Refer Slide Time: 13:30)

### Decisional Diffie-Hellman (DDH) Assumption

- ❑ **Diffie-Hellman triple** --- A triple  $(g^a, g^b, g^y)$  over  $\mathbb{G}^3$  is called a **DH-triple** if  $y = ab$
- ❑ **DDH problem** --- to **efficiently distinguish** a random DH-triple from a random triple over  $\mathbb{G}^3$

Publicly known **group description** of a cyclic group ---  $(\mathbb{G}, o, q, g): |q| = \lambda$ ,  
with  $\text{poly}(\lambda)$ -time algorithms for performing group exponentiations

Experiment:  $\text{DDH}_{\mathcal{A}, \mathbb{G}}(\lambda)$

- ❖  $a, \beta, \gamma \in_r \{0, \dots, q-1\}$
- ❖  $u \leftarrow g^a, v \leftarrow g^\beta$
- ❖  $b \in_r \{0, 1\}$
- If  $b = 0, w \leftarrow g^\gamma$
- If  $b = 1, w = g^{a\beta}$

Experiment:  $\text{DDH}_{\mathcal{A}, \mathbb{G}}(\lambda)$

$\text{DDH}_{\mathcal{A}, \mathbb{G}}(\lambda) \stackrel{\text{def}}{=} 1$ , if and only if  $b' = b$

- ❑ **Definition (DDH assumption)**: DDH assumption holds in  $(\mathbb{G}, o)$ , if for every PPT  $\mathcal{A}$ , there is a function  $\text{negl}(\lambda)$ :  
 $\Pr[\text{DDH}_{\mathcal{A}, \mathbb{G}}(\lambda) = 1] \leq 1/2 + \text{negl}(\lambda) \approx \Pr[\mathcal{A} \text{ outputs } b' = 1 | b = 1]$   
 $\quad \quad \quad - \Pr[\mathcal{A} \text{ outputs } b' = 1 | b = 0] \leq \text{negl}(\lambda)$
- ❑ Several candidate groups, where DDH assumption is **strongly believed** (but not yet proved) to be true

Now to design the oblivious transfer protocol looking ahead, we will require actually a variant of the DLog problem which we call as the decisional Diffie-Hellman or DDH problem and the corresponding assumption is the DDH assumption. In fact, we will see a variant of the DDH assumption itself which feasibly required in the oblivious transfer protocol. So what is this DDH assumption?

So first, let us define Diffie-Hellman triple. So if you are given a collection of three elements from the set  $G$  or the group  $G$ , then we call the triplet as a Diffie-Hellman triplet if the following holds. Imagine that the first element in the triplet is  $g$  power alpha. The second element in the triplet is  $g$  power beta and the third element in the triplet is  $g$  power gamma.

I can always express any element from the group as some power of the generator. That is what I have done here. So we will call this triplet as a Diffie-Hellman triplet if  $\gamma$  is equal to the product of  $\alpha$  times  $\beta$ . Otherwise, it is called a non Diffie-Hellman triple. And the DDH problem is the following. The DDH problem is to efficiently distinguish a random Diffie-Hellman triple from a random triple over the group.

And it is believed that for certain candidate cyclic groups, solving the DDH problem is indeed difficult. So to model the difficulty of the DDH problem, let us introduce the DDH experiment. This is again a game played between a hypothetical verifier and a hypothetical adversary who would like to solve the DDH problem. So the challenge for the adversary is prepared as follows.

So the verifier will first pick two random group elements and for doing that it randomly picks the indices  $\alpha$  and  $\beta$  in the range  $0$  to  $q - 1$ . And now it tosses a fair coin a small  $b$ . If the coin toss is  $0$ , then it computes a third element which is completely random namely  $g$  to the power  $\gamma$ . Whereas, if the coin toss is equal to  $1$ , then the third element that it is computing it is not a random element, but it is rather  $g$  to the power  $\alpha$  times  $\beta$ .

Now this adversary  $a$  is not aware what is the result of the coin toss whether  $b$  is equal to  $0$  or  $1$ . The challenge for the adversary is as follows. It is given the triplet  $u, v, w$ . So  $u$  is  $g$  to the power  $\alpha$ ,  $v$  is  $g$  to the power  $\beta$ . Now  $w$  is either of type  $g$  to the power  $\gamma$  where  $\gamma$  is random or it is of the form  $g$  to the power  $\alpha$  times  $\beta$ .

That means, depending upon whether the coin toss is  $0$  or  $1$  this triplet is either a non Diffie-Hellman triple or a Diffie-Hellman triple. And the challenge for the adversary is to find out what kind of triplet it is given as a challenge in polynomial amount of time. So in polynomial amount of time it has to give a response and the response has to be a bit because he has to tell what is the type of the triplet that is given, whether it is a Diffie-Hellman triple or a non Diffie-Hellman triplet.



And we say that adversary has won the experiment or the output of the experiment is 1 if and only if adversary can indeed find out what is the nature of the Diffie-Hellman triplet. Namely, his response  $b$  prime is same as the coin toss of the verifier. And the definition of the DDH assumption is the following.

We say that the DDH assumption holds in a group if for every polynomial time algorithm participating in this experiment the probability that it can find out what is the type of triplet it is given as a challenge is upper bounded by half plus some negligible function in the security parameter. Why half plus negligible?

Because there is always an adversary strategy of just guessing what exactly is the type of the triplet given to him. And the guessing strategy's success probability is upper bounded by half. We want that any polynomial time algorithm should not be able to do anything better than half okay. Another interpretation of the DDH assumption is the following.

We will say that the DDH assumption holds in the group if the distinguishing advantage of the adversary to distinguish a Diffie-Hellman triple from a non Diffie-Hellman triple is upper bounded by a negligible probability. That means it does not matter what is the type of triplet that is given as a challenge to the adversary.

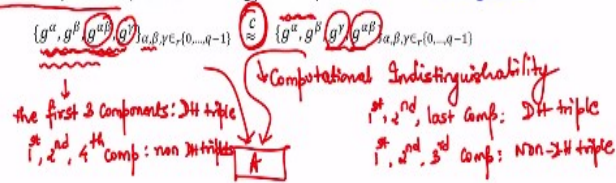
Even if it is a Diffie-Hellman triple, or even if it is a non Diffie-Hellman triple, the response of the adversary is almost identical except with some negligible probability. So if any of these two conditions hold, we say that the DDH assumption holds in the underlying group. And again, there are several candidate cyclic groups for which we strongly believe that the DDH problem is difficult to solve or the DDH assumption holds in those group.

Because as of now we do not have any polynomial time algorithm to solve the discrete to solve the DDH problem in those groups. But I stress that it is only strongly believed. That means it is only conjectured that the DDH problem is difficult to solve in those groups. We do not have any mathematical proof that indeed in this specific group, no polynomial time algorithm can solve the DDH problem, the DDH problem, okay.

(Refer Slide Time: 19:22)

## Variant of the DDH Assumption

□ Intuition: If DDH assumption holds, then the following probability distributions are indistinguishable



So finally, for our oblivious transfer protocol that we are going to see later, we are going to stick to a slight variant of the DDH assumption. And the intuition behind this variant is the following. If indeed the DDH assumption holds in my cyclic group, that means no polynomial time algorithm can solve a random instance of the DDH problem, then we can prove that the following two probability distributions are computationally indistinguishable.

So this is a notation for denoting computational indistinguishability, okay. And what is computational indistinguishability? By that we mean the following. If there is an algorithm A who is either given a sample of the left hand side type or a random sample of the right hand side type, it cannot figure out what is the type of sample it is given.

That means, it is only a negligible probability that it can distinguish apart whether it is given a sample of type 1 or the sample of type 2. That means in both the cases his response will be same, okay. That is what we mean by computational indistinguishability. So what are these two probability distributions? In both the probability distributions, we are talking about a collection of four elements from the group.

The first two components of both the distributions are some random group elements  $g$  to the power  $\alpha$ ,  $g$  to the power  $\beta$  where  $\alpha$  and  $\beta$  are randomly picked

from the set in the index range 0 to  $q - 1$ . So if the indices  $\alpha$  and  $\beta$  are randomly picked, and so then the elements  $g$  to the power  $\alpha$  and  $g$  to the power  $\beta$  will also be random elements from the group.

So the first two components for the triplets in both the probability distributions are identically distributed, okay.  $g$  to the power  $\alpha$  is a random element,  $e$  to the power  $\alpha$  is a random element in both the distributions. The second component is a random group element  $g$  to the power  $\beta$  in the first two distribution, the second component is a random element in the second probability distribution.

The difference is in the type of the third and the fourth component. In the first probability distribution, the third component is  $g$  to the power  $\alpha, \beta$ . Whereas the third component in the second probability distribution is a totally random group element.

Whereas, if I compare the fourth component of the two probability distributions, then in the first probability distribution, the fourth element of the this collection of four elements, the fourth element is a random element whereas the fourth element in the second probability distribution is a random group element. So what we can say is the following.

In the first probability distribution, the first three components is a Diffie-Hellman triple and the first, second and fourth component is a non Diffie-Hellman. That is the nature of the, nature of the values in the first probability distribution. Whereas, if I consider the second probability distribution, then the first, second, last component together they constitute a Diffie-Hellman triple.

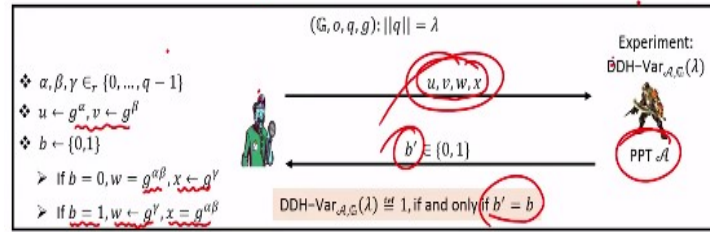
Whereas, the first, second, third component they constitute a non Diffie-Hellman triple, okay. And the variant of this DDH assumption is that, if we assume that the DDH assumption holds in my underlying group or the DDH problem is difficult to solve in the underlying group, then indeed these two probability distributions are computationally indistinguishable.

**(Refer Slide Time: 24:05)**

## Variant of the DDH Assumption

□ Intuition: If DDH assumption holds, then the following probability distributions are indistinguishable

$$\{g^a, g^b, g^{ab}, g^y\}_{a,b,y \in \{0, \dots, q-1\}} \stackrel{c}{\approx} \{g^a, g^b, g^y, g^{ab}\}_{a,b,y \in \{0, \dots, q-1\}}$$



□ **Definition:** DDH-Var assumption holds in  $(\mathbb{G}, o)$ , if for every PPT  $\mathcal{A}$ , there is a function  $\text{negl}(\lambda)$ :

$$\Pr[\text{DDH-Var}_{A,G}(\lambda) = 1] \leq \frac{1}{2} + \text{negl}(\lambda) \approx |\Pr[\mathcal{A} \text{ outputs } b' = 1 | b = 1] - \Pr[\mathcal{A} \text{ outputs } b' = 1 | b = 0]| \leq \text{negl}(\lambda)$$

□ **Claim:** If DDH assumption holds in  $(\mathbb{G}, o)$  then DDH-Var assumption also holds in  $(\mathbb{G}, o)$

And this is formalized again by an experiment which is just a slight variant of the DDH experiment. Why slight variant because now we have to give a challenge to the adversary which has four elements. Because we are now talking about computational indistinguishability of two probability distributions, each of which has four components, okay. So the challenge for the adversary is as follows.

It is given four elements from the group. How these four elements are generated or how this challenge has been prepared for the adversary? So the first two elements  $u$  and  $v$ , they are random group elements. And then this verifier tosses a coin. If the coin toss is 0, then the third component along with the first two components constituted a Diffie-Hellman triple.

And the fourth component along with the first two components constitute a non Diffie-Hellman triple. Whereas if the coin toss is  $b$  equal to 1, then the third component along with the first two components constitute a non Diffie-Hellman triple. And the last component along with the first two components constitute a Diffie-Hellman triple.

And the challenge for this adversary is to find out what exactly is the type of challenge it is seeing. Whether it is seeing a challenge of type  $b = 0$  or a challenge of type  $b = 1$  in polynomial amount of time. So in polynomial amount of time, he has to come up with an answer. And we will say that his answer is correct or equivalently he has won the experiment if and only if  $b$  prime is equal to  $b$ .

And we will say that this variant of the DDH assumption holds in the underlying group if for every polynomial time algorithm participating in this experiment, the probability that he can win the experiment or can find out the type of challenge that is given to him is upper bounded by some half plus negligible probability in the security parameter. Or equivalently the distinguishing advantage of the adversary is negligible.

That means it does not matter what is the type of challenge it is given, type of challenge given to the adversary. Whether it is of type  $b = 1$  or  $b = 0$ , the response of the adversary is almost identical in both cases. And we can formally prove that if the DDH assumption holds in your underlying group.

That means if indeed your candidate cyclic group is such that the DDH problem is very difficult to solve in polynomial amount of time then even this variant of the DDH problem is also difficult to solve in the same group. That means in polynomial amount of time, no adversary or no algorithm can find out what is the type of challenge that is given to him, okay.

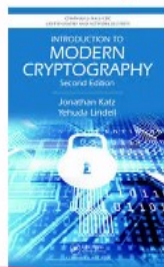
And as I said, looking ahead, we will use this variant of the DDH assumption while designing our oblivious transfer protocols for strings.

**(Refer Slide Time: 27:15)**

---

## References

- For cyclic group, DLog and DDH assumption



<https://nptel.ac.in/courses/106/106/106106221/>

- For the variant of DDH assumption and relation with DDH assumption



---

So these are the references which are used for today's lecture. As I said, I quickly went through the concepts related to cyclic groups, DLog, DDH assumptions. All of

them are taken from this book by Katz and Lindell. Or if you want the video lectures, you are referred to this NPTEL course on foundations of cryptography.

And for the variant of the DDH assumption that I discussed towards the end, you are referred to this textbook on Efficient Secure Two-Party Protocols by Hazay and Lindell. Thank you.