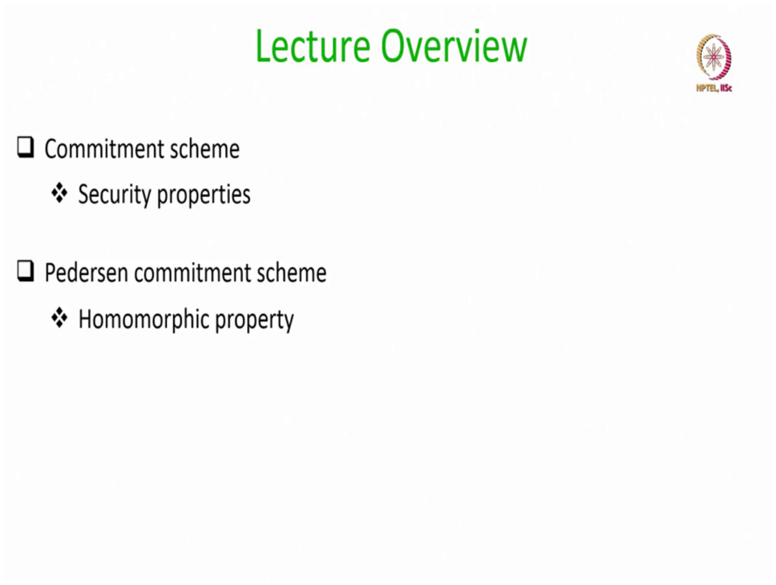
Secure Computation: Part II
Prof. Ashish Choudhury
Department of Computer Science and Engineering
Indian Institute of Science, Bengaluru

Lecture - 60
Pedersen Commitment Scheme

(Refer Slide Time: 00:24)

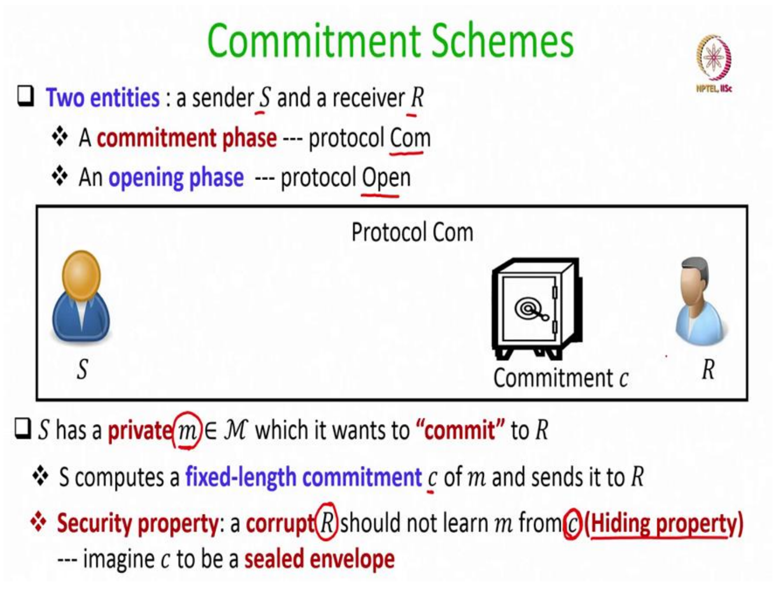


Lecture Overview

- ❑ Commitment scheme
  - ❖ Security properties
- ❑ Pedersen commitment scheme
  - ❖ Homomorphic property

Hello everyone, welcome to this lecture. So, in this lecture we will discuss about commitment schemes as a primitive and then we will see an instantiation of commitment scheme, namely Pedersen's commitment scheme and its homomorphic property.

(Refer Slide Time: 00:39)



Commitment Schemes

- Two entities : a sender S and a receiver R
  - A commitment phase --- protocol Com
  - An opening phase --- protocol Open

Protocol Com
S
Commitment c
R

- S has a private $m \in \mathcal{M}$ which it wants to "commit" to R
  - S computes a fixed-length commitment c of m and sends it to R
  - Security property: a corrupt R should not learn m from c (Hiding property) --- imagine c to be a sealed envelope

So, a commitment scheme is a very important cryptographic primitive, it involves two entities a sender and a receiver. And a commitment scheme will have two phases, each phase implemented by a separate protocol.

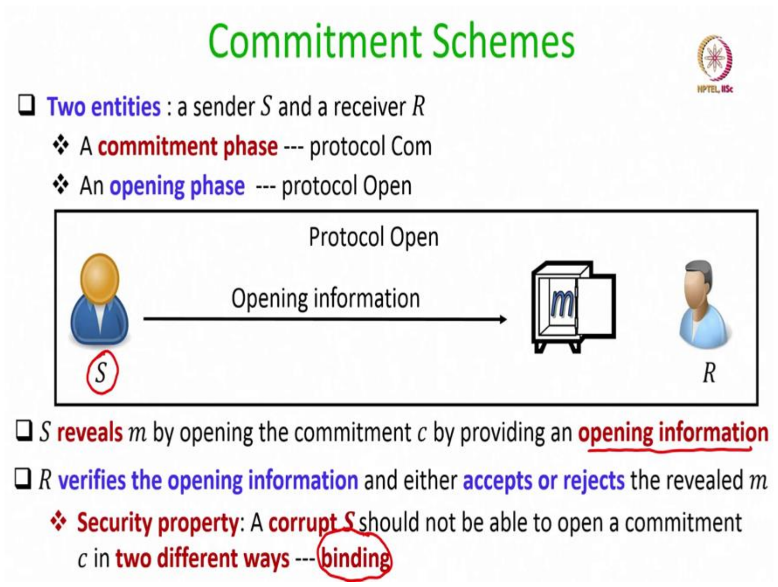
So, we have a commitment phase implemented by a protocol Com and we have an opening phase implemented by the protocol Open. In the commitment phase the sender will have some input m from some domain which it would like to commit to R. So, to do that it, will compute some commitment as per the com protocol, the commitment will be denoted by c and this commitment will be given to the receiver.

So, the security property that we require here from this commitment phase is the following. If the receiver is corrupt, then by seeing the commitment c it should not learn anything regarding the value which has been committed by the S. Namely, it should not learn anything about the value of m, this is called as the hiding property. So, you can imagine the commitment c to be some kind of sealed envelope and the sealed envelope is given to this receiver and receiver should not be able to find out what exactly is kept inside the envelope, that is the analogy here.

I would like to stress here that the hiding property I have discussed here is very loose, it is not very formal. To formalize that the receiver does not learn anything about the message m from the commitment c, we can use the notion of indistinguishability. Loosely speaking that demands that even if receiver gives a pair of messages m0 and m1 to the sender and if sender randomly commits one of those messages, receiver should not be able to tell whether it has seen the commitment of m0 or it has seen the commitment of m1.

But for the purpose of understanding, for the ease of understanding, we will loosely say that the commitment hiding property means that the commitment should not reveal anything about the sender's message to the receiver.

(Refer Slide Time: 03:01)



Commitment Schemes

- **Two entities** : a sender S and a receiver R
  - A **commitment phase** --- protocol Com
  - An **opening phase** --- protocol Open

Protocol Open
Opening information
m
S
R

- S **reveals** m by opening the commitment c by providing an **opening information**
- R **verifies the opening information** and either **accepts or rejects** the revealed m
  - **Security property**: A **corrupt** S should not be able to open a commitment c in **two different ways** --- **binding**

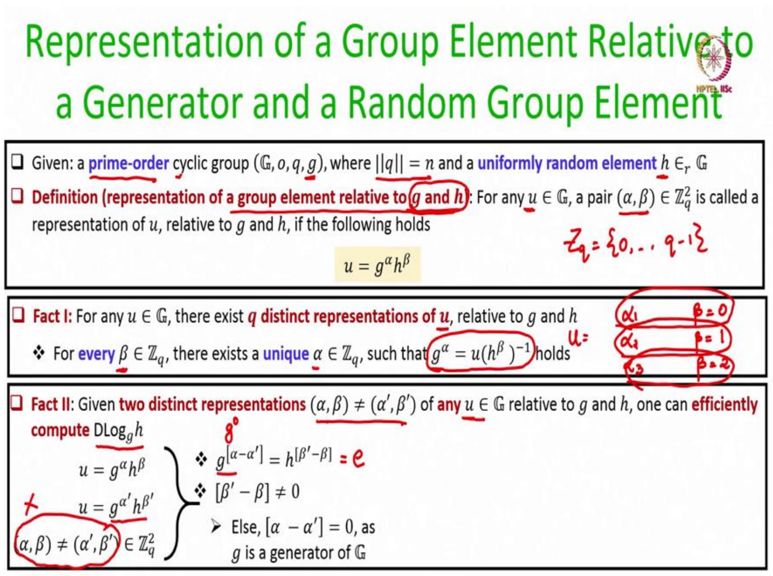
So, that is the commitment phase, where the sealed envelope is given to the receiver namely a value has been committed. Now, the opening phase is implemented by the open protocol, where S would now like to reveal the value which had which it has committed in the commitment c.

So, the way this commitment phase and opening phase are used in a primitive will be as follows. So, there will be always some gap between the commitment phase and an opening phase, there might be a scenario where sender would like to commit some values and once the values has been committed, we would like to open and check what is the value which had been committed.

So, to check what value has been committed, the Open protocol will be used where S has to reveal the committed value by providing some kind of opening information. Now, using this opening information receiver will open the commitment and after opening it either accepts or reject the revealed value.

The security property which we require here is the binding property, which informally requires that a corrupt S should be committed to the value which it has committed earlier in the commitment phase; that means, it should not be possible for a corrupt sender to commit to m during the commitment phase, but later he is able to open the commitment to m star, where m star is different from m, that should not happen. Again this requirement can be modeled through some security experiment, but I am not going into the details of those security experiment.

(Refer Slide Time: 04:51)



Representation of a Group Element Relative to a Generator and a Random Group Element

- Given: a **prime-order** cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a **uniformly random element** $h \in_r \mathbb{G}$
- **Definition (representation of a group element relative to g and h)**: For any $u \in \mathbb{G}$, a pair $(\alpha, \beta) \in \mathbb{Z}_q^2$ is called a representation of u, relative to g and h, if the following holds

$$u = g^{\alpha} h^{\beta}$$

$\mathbb{Z}_q = \{0, .. , q-1\}$

- **Fact I:** For any $u \in \mathbb{G}$, there exist **q distinct representations of u**, relative to g and h
  - For **every** $\beta \in \mathbb{Z}_q$, there exists a **unique** $\alpha \in \mathbb{Z}_q$, such that $g^{\alpha} = u(h^{\beta})^{-1}$ holds

$u =$ α_1 $\beta = 0$ α_1 $\beta = 1$ α_3 $\beta = 2$

- **Fact II**: Given **two distinct representations** $(\alpha, \beta) \neq (\alpha', \beta')$ of **any** $u \in \mathbb{G}$ relative to g and h, one can **efficiently compute** $\text{DLog}_g h$

$$\left.\begin{array}{l} u = g^{\alpha} h^{\beta} \\ u = g^{\alpha'} h^{\beta'} \\ (\alpha, \beta) \neq (\alpha', \beta') \in \mathbb{Z}_q^2 \end{array}\right\}$$

  - $g^{[\alpha - \alpha']} = h^{[\beta' - \beta]} = e$ (g^0)
  - $[\beta' - \beta] \neq 0$
    - Else, $[\alpha - \alpha'] = 0$, as g is a generator of $\mathbb{G}$

So, there are several instantiations of commitment schemes, for the purpose of our cryptographically secure verifiable secret sharing, we will be using Pedersen's commitment scheme.

To understand the Pedersen's commitment scheme, we will first try to understand the representation of a group element with respect to a generator and a random element. So, imagine you are given a prime order cyclic group, where the size of the group is exponentially large in n and you are also given a uniformly random group element h ok. So now, you are given the generator as well as a random element h from the group and a discrete logarithm of this element h is not known ok.

Now, for any group element u, we say that a pair of indices (α, β) from the set Zq is a representation of u, if u is equal to $g^{\alpha} h^{\beta}$, ok that is the definition of a representation of a group element relative to g and h. So, the elements g and h are fixed, you can treat them as some kind of base with respect to that base, (α, β) will be considered as a representation for u, if $g^{\alpha}h^{\beta}$ gives you the element u.

Now, you might be wondering whether there exists a unique representation for any given group element, the answer is no. Because if I give you any element u there could be up to q number of distinct representations with respect to the fixed base g and h. This is because, if I take any candidate beta as one part of the representation, then corresponding to that beta, I can always find the corresponding unique alpha, such that this relationship holds.

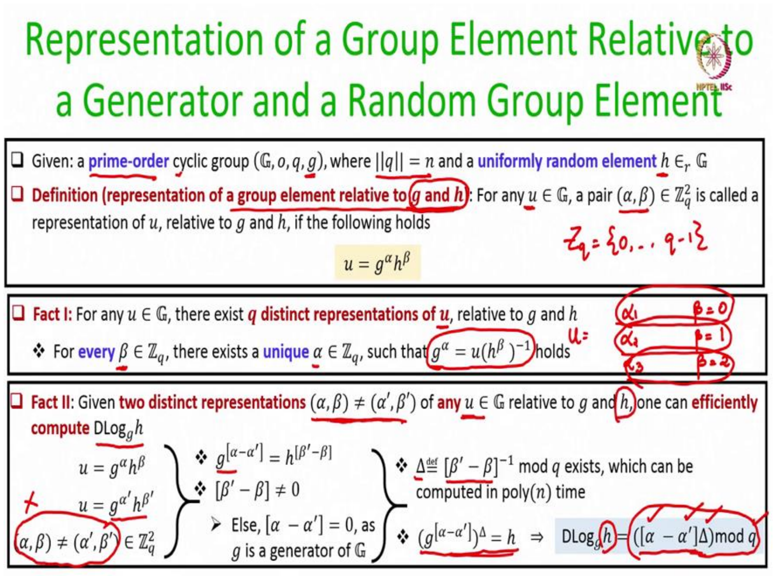
Now, how many betas I can have? I can have q number of betas, because remember both alpha and beta are elements from Zq and Zq is nothing but the 0 to q minus 1. So that means, if you set beta equal to 0, corresponding to that there will be some alpha, if you set beta equal to 1 then corresponding to that there will be some another alpha. If you set beta equal to 2 corresponding to that there will be some another alpha and so on, such that all of them constitute a representation of the same element u; that means, g to the power alpha 1, h to the power 0 will give you u, g to the power alpha 2 times h to the power 1 will give you u, g to the power alpha 3 times h square will give u and so on. So, with respect to this fixed base g and h, there could be up to q number of representations. Another fact which we can quickly derive here is the following.

If you are given a pair of distinct representations for any element u in the group, with respect to the base g and h then using these two representations you can easily compute the discrete logarithm of h. How? So, since alpha, beta is a representation of u; that means, u is equal to g to the power alpha times h to the power beta. And since alpha prime, beta prime is also a representation of u, you have g to the power alpha prime times h to the power beta prime also giving you u. And remember that we are assuming here that the representations alpha, beta and alpha prime, beta prime are different.

Now, if this is the case then I can say that g to the power alpha minus alpha prime is same as h to the power beta prime minus beta and this automatically implies that the difference of beta prime and beta has to be non-zero, because if the difference of beta prime and beta is 0, then the right-hand side here becomes h to the power 0, which is the identity element. That means, we have g to the power alpha minus alpha prime giving you the identity element; that means, it is g to the power 0 here the left-hand side; that means, alpha minus alpha prime is 0.

And alpha minus alpha prime being 0 means alpha is equal to alpha prime and beta prime minus beta being 0 means beta prime equal to beta, but that is against this assumption that the representation alpha, beta and alpha prime, beta prime are distinct.

(Refer Slide Time: 10:18)



Representation of a Group Element Relative to a Generator and a Random Group Element

- Given: a **prime-order** cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a **uniformly random element** $h \in_r \mathbb{G}$
- **Definition (representation of a group element relative to g and h)**: For any $u \in \mathbb{G}$, a pair $(\alpha, \beta) \in \mathbb{Z}_q^2$ is called a representation of u, relative to g and h, if the following holds

$$u = g^{\alpha} h^{\beta}$$

$\mathbb{Z}_q = \{0, .. , q-1\}$

- **Fact I:** For any $u \in \mathbb{G}$, there exist **q distinct representations of u**, relative to g and h
  - For **every** $\beta \in \mathbb{Z}_q$, there exists a **unique** $\alpha \in \mathbb{Z}_q$, such that $g^{\alpha} = u(h^{\beta})^{-1}$ holds

$u =$ α_1 $\beta = 0$, α_2 $\beta = 1$, α_3 $\beta = 2$

- **Fact II**: Given **two distinct representations** $(\alpha, \beta) \neq (\alpha', \beta')$ of **any** $u \in \mathbb{G}$ relative to g and h, one can **efficiently compute** $\text{DLog}_g h$

$u = g^{\alpha} h^{\beta}$

$u = g^{\alpha'} h^{\beta'}$

$(\alpha, \beta) \neq (\alpha', \beta') \in \mathbb{Z}_q^2$

- $g^{[\alpha-\alpha']} = h^{[\beta'-\beta]}$
- $[\beta' - \beta] \neq 0$
  - Else, $[\alpha - \alpha'] = 0$, as g is a generator of $\mathbb{G}$

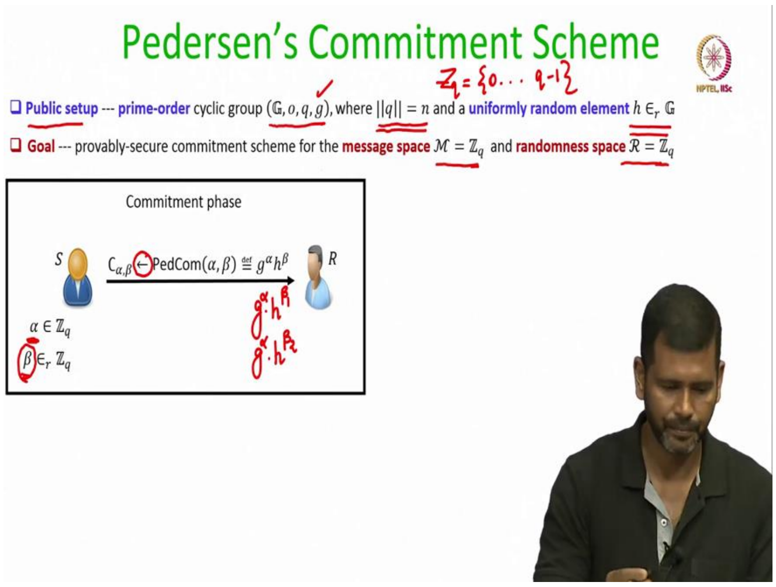
- $\Delta \overset{\text{def}}{=} [\beta' - \beta]^{-1} \bmod q$ exists, which can be computed in poly(n) time
- $(g^{[\alpha-\alpha']})^{\Delta} = h \Rightarrow \text{DLog}_g h = ([\alpha - \alpha']\Delta) \bmod q$

So; that means, the right-hand side is definitely not h to the power 0 here; that means, the inverse of beta prime minus beta exists because it is a non 0 element. And we can easily find out that inverse in polynomial time, say by using the extended Euclid algorithm.

So, let me denote the inverse of beta prime minus beta by delta. As I said, this can be computed in polynomial time. So, now, given alpha, beta, alpha prime, beta prime and u, it is easy to see that this relationship holds and this implies that the discrete logarithm of h is nothing but the value alpha minus alpha prime times delta. And if this value is beyond q minus 1, you can bring it within the range 0 to q minus 1 by performing mod q.

So, you are given alpha, you are given alpha prime, you are given delta, you are given q by computing this value, you will be able to compute discrete logarithm of h. So that means, the fact two shows here that if you know a way to come up with two distinct representations for any group element u, then using the two distinct representations you can solve an instance of discrete logarithm problem. Namely, you can compute a discrete logarithm of this random group element h in polynomial amount of time without doing any kind of brute force operations.

(Refer Slide Time: 12:04)



Pedersen's Commitment Scheme

$\mathbb{Z}_q = \{0 \dots q-1\}$

- Public setup --- prime-order cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a uniformly random element $h \in_r \mathbb{G}$
- Goal --- provably-secure commitment scheme for the message space $\mathcal{M} = \mathbb{Z}_q$ and randomness space $\mathcal{R} = \mathbb{Z}_q$

Commitment phase

S $\xrightarrow{C_{\alpha,\beta} \leftarrow \text{PedCom}(\alpha,\beta) \overset{\text{def}}{=} g^{\alpha}h^{\beta}}$ R

$\alpha \in \mathbb{Z}_q$

$\beta \in_r \mathbb{Z}_q$

$g^{\alpha} \cdot h^{\beta_1}$

$g^{\alpha} \cdot h^{\beta_2}$

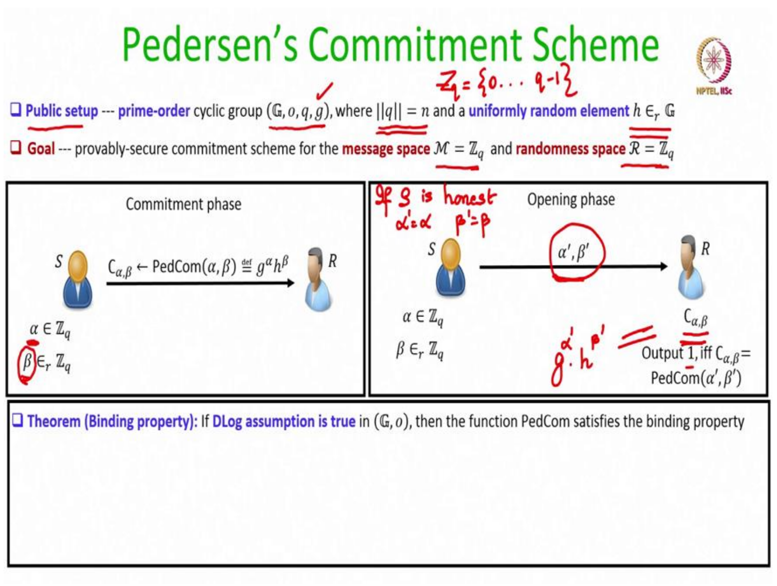
Now, based on all these things, let us see the Pedersen's commitment scheme, there will be a public setup which will be used in the scheme, the public setup is that of a prime order cyclic group. Since, it is a cyclic group, a generator will be given to us and the order of the cyclic group is exponentially large in this parameter n. As part of the setup you will be also provided with a random group element h, the message space for this commitment scheme will be the set Zq.

So, let me write down what is Zq here, the set Zq will have the elements 0 to q minus 1; that means, this commitment scheme can allow the sender to commit any value from this set Zq, we will be using some randomness as part of the commitment scheme, the randomness space also will be the set Zq. The commitment phase is as follows: so, the com algorithm will do the following operations, if there is a value alpha, I should have used M to represent the input of S. But I am using alpha to represent the value which the sender would like to commit. So, to commit the value alpha, it picks a randomness beta which is also a random element from the randomness space and the commitment is g to the power alpha times h to the power beta, which is denoted by C subscript alpha, beta. I am using the left arrow here to represent the output of the Pedersen commitment algorithm.

Because this commitment algorithm is a randomized algorithm; that means, every time sender would like to commit the same value alpha, the commitment output, commitment will be different because beta will be picked randomly for every instance. So, first time it might be g to the power alpha times h to the power beta 1, next time it will be g to the power alpha times h to the power beta 2 and so on.

That means if the same value alpha is committed multiple times, a corrupt receiver cannot simply say that the two committed values are same or they are different and so on ok, by comparing the commitments.

(Refer Slide Time: 14:40)



Pedersen's Commitment Scheme

$\mathbb{Z}_q = \{0 \ldots q-1\}$

- Public setup --- prime-order cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a uniformly random element $h \in_r \mathbb{G}$
- Goal --- provably-secure commitment scheme for the message space $\mathcal{M} = \mathbb{Z}_q$ and randomness space $\mathcal{R} = \mathbb{Z}_q$

Commitment phase
S
$C_{\alpha,\beta} \leftarrow \text{PedCom}(\alpha, \beta) \overset{\text{def}}{=} g^{\alpha} h^{\beta}$
R
$\alpha \in \mathbb{Z}_q$
$\beta \in_r \mathbb{Z}_q$
If S is honest
$\alpha' = \alpha$ $\beta' = \beta$
Opening phase
S
α', β'
R
$\alpha \in \mathbb{Z}_q$
$\beta \in_r \mathbb{Z}_q$
$C_{\alpha,\beta}$
$g^{\alpha'} \cdot h^{\beta'} =$
Output 1, iff $C_{\alpha,\beta} = \text{PedCom}(\alpha', \beta')$

- Theorem (Binding property): If DLog assumption is true in $(\mathbb{G}, o)$, then the function PedCom satisfies the binding property

So, that is the commitment phase. And the opening phase is very simple. If sender wants to reveal the value which it had committed earlier, then it simply gives in clear the value committed and the randomness used.

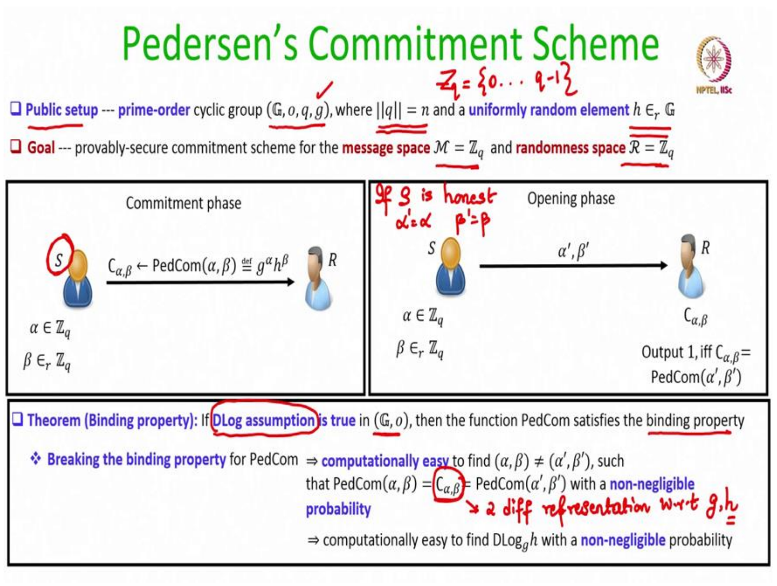
So, if sender is corrupt, it might try to give now a different value for alpha and the randomness. So, that is why I am using the notation alpha prime and beta prime, if sender is honest, if S is honest, then alpha prime will be same as alpha and beta prime will be same as beta ok, but if the sender is corrupt then this need not be the case.

Now, how does the receiver verify the opening information? So, it already has the commitment, C of alpha, beta. And now it is seeing the values revealed by the sender, it itself recomputes the commitment of alpha prime with respect to the randomness beta prime, namely receiver itself recomputes g to the power alpha prime h to the power beta prime and it checks whether it matches the commitment which it had received earlier.

If it matches then the output is 1, that means accept, otherwise the output is 0, namely reject. So, now, let us see whether the hiding and the binding properties are achieved by this commitment scheme. So, let us first argue about the binding property and remember for binding we have to consider a potentially corrupt sender.

So, the claim here is that if the discrete log assumption is true in the group, that means in polynomial amount of time, it is not possible to solve an instance random instance of discrete log except with negligible probability, then the commitment scheme satisfies the binding property.

(Refer Slide Time: 16:43)



Pedersen's Commitment Scheme

$\mathbb{Z}_q = \{0 \dots q-1\}$

- Public setup --- prime-order cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a uniformly random element $h \in_r \mathbb{G}$
- Goal --- provably-secure commitment scheme for the message space $\mathcal{M} = \mathbb{Z}_q$ and randomness space $\mathcal{R} = \mathbb{Z}_q$

Commitment phase

S: $\alpha \in \mathbb{Z}_q$, $\beta \in_r \mathbb{Z}_q$

$C_{\alpha,\beta} \leftarrow \text{PedCom}(\alpha, \beta) \overset{\text{def}}{=} g^\alpha h^\beta$ → R

Opening phase

If S is honest $\alpha' = \alpha$ $\beta' = \beta$

S: $\alpha \in \mathbb{Z}_q$, $\beta \in_r \mathbb{Z}_q$

α', β' → R: $C_{\alpha,\beta}$

Output 1, iff $C_{\alpha,\beta} = \text{PedCom}(\alpha', \beta')$

- Theorem (Binding property): If DLog assumption is true in $(\mathbb{G}, o)$, then the function PedCom satisfies the binding property
  - Breaking the binding property for PedCom $\Rightarrow$ computationally easy to find $(\alpha, \beta) \neq (\alpha', \beta')$, such that $\text{PedCom}(\alpha, \beta) = C_{\alpha,\beta} = \text{PedCom}(\alpha', \beta')$ with a non-negligible probability
    → 2 diff representation w.r.t g, h
    $\Rightarrow$ computationally easy to find $\text{DLog}_g h$ with a non-negligible probability

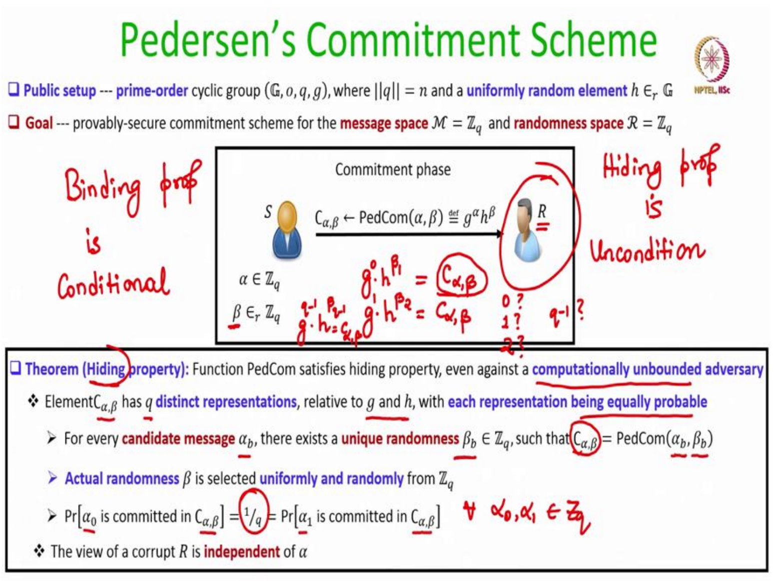
Because breaking the binding property would require the sender to come up with the pairs alpha, beta and alpha prime, beta prime which are different such that the commitment of both alpha, beta as well as alpha prime, beta prime turns out to be this common group element C of alpha, beta.

That means to break the binding property, a corrupt sender has to come up with two different representations for this element C of alpha, beta, so two different representations with respect to g and h. But recall that in the previous slide we had seen that if you can come up with two distinct representations of any group element with respect to this generator g and a random element h, then that is equivalent to solving or computing the discrete logarithm of the random element h in polynomial time. But that goes against the assumption that discrete logarithm is difficult to solve in my group.

So that means, if my sender is corrupt and its running time is polynomially bounded then except with a negligible probability, it cannot break the binding property. Of course, if sender is computationally unbounded, then it can brute force over all candidate alpha, beta and it can always come up with two distinct representations for this commitment.

So, during the commit phase, it can use alpha, beta, but during the opening phase it can use alpha prime, beta prime and receiver will accept it. But that will require the sender to perform exponential amount of computation, the binding property holds only if the sender is corrupt and its running time is computationally bounded.

(Refer Slide Time: 18:55)



Pedersen's Commitment Scheme

- **Public setup** --- **prime-order** cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a **uniformly random element** $h \in_r \mathbb{G}$
- **Goal** --- provably-secure commitment scheme for the **message space** $\mathcal{M} = \mathbb{Z}_q$ and **randomness space** $\mathcal{R} = \mathbb{Z}_q$

Commitment phase

$S \quad C_{\alpha,\beta} \leftarrow \text{PedCom}(\alpha, \beta) \overset{\text{def}}{=} g^{\alpha} h^{\beta} \longrightarrow R$

$\alpha \in \mathbb{Z}_q$

$\beta \in_r \mathbb{Z}_q$

Binding prop is Conditional
Hiding prop is Uncondition
$g^0 \cdot h^{\beta_1} = C_{\alpha,\beta}$
$g^1 \cdot h^{\beta_2} = C_{\alpha,\beta}$
$g^{q-1} \cdot h^{\beta_{q-1}} = C_{\alpha,\beta}$
0 ? 1 ? 2 ?
q-1 ?

- **Theorem (Hiding property):** Function PedCom satisfies hiding property, even against a **computationally unbounded adversary**
  - Element $C_{\alpha,\beta}$ has q **distinct representations**, relative to g and h, with **each representation being equally probable**
    - For every **candidate message** α_b, there exists a **unique randomness** $\beta_b \in \mathbb{Z}_q$, such that $C_{\alpha,\beta} = \text{PedCom}(\alpha_b, \beta_b)$
    - **Actual randomness** β is selected **uniformly and randomly** from $\mathbb{Z}_q$
    - $\Pr[\alpha_0 \text{ is committed in } C_{\alpha,\beta}] = 1/q = \Pr[\alpha_1 \text{ is committed in } C_{\alpha,\beta}] \quad \forall\, \alpha_0, \alpha_1 \in \mathbb{Z}_q$
  - The view of a corrupt R is **independent** of α

Now, let us see the hiding property for which we have to consider a corrupt receiver. And interestingly now we can claim here that even if the receiver is computationally unbounded the hiding property will hold.

So, this is quite asymmetric. The binding property is conditional because it requires the sender to be restricted to perform only polynomial amount of computation or the adversary to perform only polynomial amount of computation. But the hiding property here is unconditional; that means, if the receiver is corrupt and even if it is allowed unbounded resources, unbounded time, it cannot figure out what is the value which had been committed ok.

So, a natural question will be can we have a commitment scheme, where the binding as well as the hiding property are both unconditional and the answer is no. Either of these two properties has to be conditional. Now, coming back to Pedersen's commitment scheme let us see why the hiding property here is unconditional. So, let us imagine a corrupt receiver. So, what exactly the receiver is seeing in the commitment phase, it is seeing this commitment C of alpha, beta which is a group element.

Now, remember fact 1 states that this commitment will have q distinct representations with respect to the generator g and the random element h, with each representation being equally probable, from the viewpoint of the receiver.

Namely, if the receiver thinks in its mind that it is the value alpha sub b which has been committed by the sender, then corresponding to that candidate alpha sub b there is indeed a randomness, a unique randomness, say beta sub b from the randomness space, such that the commitment of alpha sub b with respect to the randomness beta sub b will give you the commitment which receiver has seen.

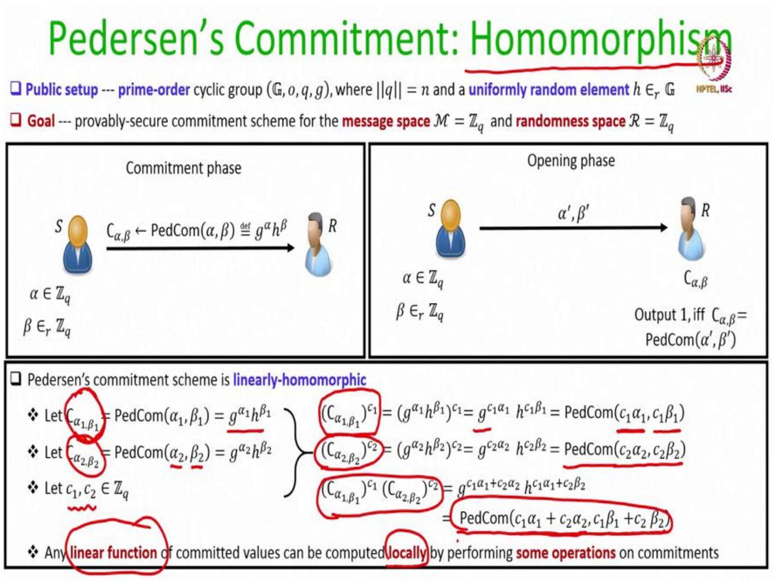
But what is the probability that the randomness used by the sender is indeed beta sub b? Well, that is 1 over q, because the randomness used in the protocol is uniformly picked from the randomness space. So that means, for all candidate alpha 0, alpha 1, from Z q the probability that alpha 0 could have been committed in this commitment C alpha, beta is 1 over q.

And the probability that alpha 1 would have been committed in this same commitment C alpha, beta is also 1 over q. So that means, just by seeing the value C alpha beta, this receiver cannot figure out whether it is the value 0 which is committed or whether it is the value 1 which is committed or whether it is the value 2 which is committed or whether it is the value q minus 1 which is committed.

With equal probability, it could be the case that sender has computed g to the power 0 times h to the power some beta 1, giving you the commitment. With equal probability it could be the case that sender has computed g to the power 1 times h to the power some beta 2 giving you the same C of alpha, beta. And with equal probability it is the case that sender has computed g to the power q minus 1 times h to the power some beta q minus 1, giving you the same commitment C of alpha, beta.

So, receiver cannot pinpoint whether sender's input was 0 or 1 or 2 or q minus 1. That means, the commitment which it sees, its probability distribution is independent of the input of the sender and this hold even if the receiver is computationally unbounded and this implies that the hiding property holds here ok.

(Refer Slide Time: 23:50)



Pedersen's Commitment: Homomorphism

- **Public setup** --- **prime-order** cyclic group $(\mathbb{G}, o, q, g)$, where $||q|| = n$ and a **uniformly random element** $h \in_r \mathbb{G}$
- **Goal** --- provably-secure commitment scheme for the **message space** $\mathcal{M} = \mathbb{Z}_q$ and **randomness space** $\mathcal{R} = \mathbb{Z}_q$

Commitment phase
S
$C_{\alpha,\beta} \leftarrow \text{PedCom}(\alpha, \beta) \overset{\text{def}}{=} g^{\alpha} h^{\beta}$
R
$\alpha \in \mathbb{Z}_q$
$\beta \in_r \mathbb{Z}_q$
Opening phase
S
α', β'
R
$\alpha \in \mathbb{Z}_q$
$\beta \in_r \mathbb{Z}_q$
$C_{\alpha,\beta}$
Output 1, iff $C_{\alpha,\beta} = \text{PedCom}(\alpha', \beta')$

- Pedersen's commitment scheme is **linearly-homomorphic**
  - Let $C_{\alpha_1,\beta_1} = \text{PedCom}(\alpha_1, \beta_1) = g^{\alpha_1} h^{\beta_1}$
  - Let $C_{\alpha_2,\beta_2} = \text{PedCom}(\alpha_2, \beta_2) = g^{\alpha_2} h^{\beta_2}$
  - Let $c_1, c_2 \in \mathbb{Z}_q$

$$(C_{\alpha_1,\beta_1})^{c_1} = (g^{\alpha_1} h^{\beta_1})^{c_1} = g^{c_1\alpha_1} \; h^{c_1\beta_1} = \text{PedCom}(c_1\alpha_1, c_1\beta_1)$$

$$(C_{\alpha_2,\beta_2})^{c_2} = (g^{\alpha_2} h^{\beta_2})^{c_2} = g^{c_2\alpha_2} \; h^{c_2\beta_2} = \text{PedCom}(c_2\alpha_2, c_2\beta_2)$$

$$(C_{\alpha_1,\beta_1})^{c_1} (C_{\alpha_2,\beta_2})^{c_2} = g^{c_1\alpha_1 + c_2\alpha_2} \; h^{c_1\alpha_1 + c_2\beta_2} = \text{PedCom}(c_1\alpha_1 + c_2\alpha_2, c_1\beta_1 + c_2\beta_2)$$

  - Any **linear function** of committed values can be computed locally by performing **some operations** on commitments

So, that is a very simple Pedersen's commitment scheme. Now, a very interesting property of the Pedersen's commitment scheme is its associated homomorphic property.

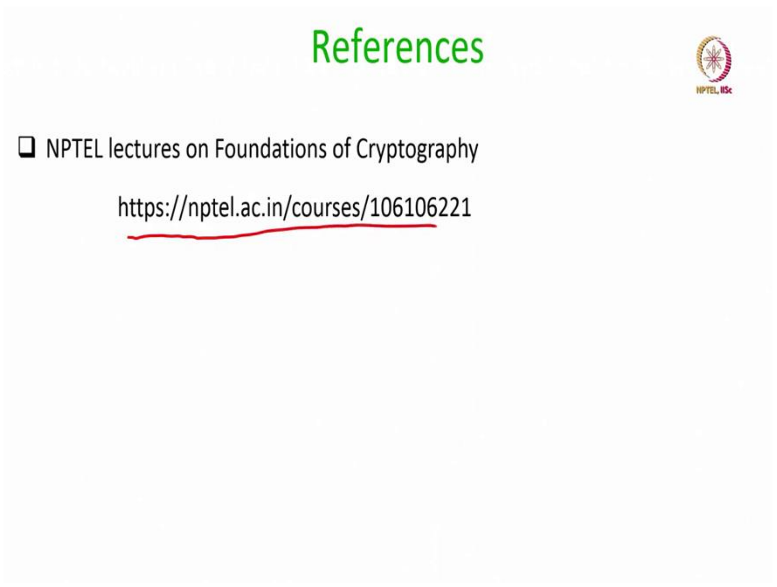
So, the Pedersen's commitment scheme is linearly homomorphic. To understand that, imagine you are given the commitment of the value alpha 1, with respect to the randomness beta 1. And say there is another value alpha 2 which is committed with respect to the randomness beta 2 ok. You also imagine that you are given two values c1, c2, treat them as constants from the message space. Now, what will happen if I take the first commitment and raise it to the power c1. Mind it, this operation, the first commitment raised to the power c1, can be performed in polynomial amount of time, because this can be done using the square and multiply method, because this is nothing but the group exponentiation operation. So, if I raise the first commitment to the power c1, that is nothing but the Pedersen commitment of the value c1 times alpha 1 with respect to the randomness c1 times beta 1.

And in the same way if I take the second commitment and raise it to the power c2, that will give me the commitment of c2 times alpha 2 with respect to the randomness c2 times beta 2. That means, now if I take these two new values and multiply them and then if I rearrange the terms, I can see that this final thing which I have computed here is nothing but a Pedersen commitment of c1 times alpha 1 plus c2 times alpha 2, with respect to the randomness c1 times beta 1 plus c2 times beta 2.

That means if someone gives me the commitment of alpha 1 and the commitment of alpha 2 and now if I would like to have a commitment of c1 times alpha 1 plus c2 times alpha 2, I do not require the sender again to freshly compute a commitment of this value c1 times alpha 1 plus c2 times alpha 2 and give it to me. Just by performing some operations on the existing commitments of alpha 1 and alpha 2, I will end up getting a commitment of c1 times alpha 1 plus c2 times alpha 2.

In general, that implies that if you have several committed values available and if you want to compute a linear function of those committed values, you can do that, it does not require any interaction whatsoever. You just perform some operations on the commitments which are already given to you, that will give you the commitment of the resultant output of this linear function. Looking ahead, we will exploit this homomorphic property when we design cryptographically secure verifiable secret sharing scheme.

(Refer Slide Time: 27:16)



References

- NPTEL lectures on Foundations of Cryptography

  https://nptel.ac.in/courses/106106221

So, these are the references. So, you can find more about commitments schemes from this lecture series.

Thank you.