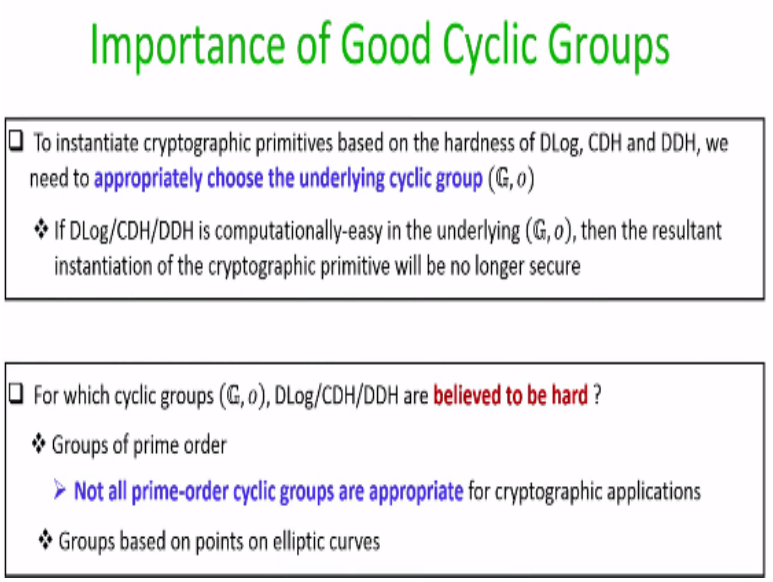
Foundations of Cryptography
Dr. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bangalore

Lecture - 40
Candidate Cyclic Groups for Cryptographic Purposes Part I

(Refer Slide Time: 00:27)

Hello everyone. Welcome to this lecture. So the plan for this lecture is as follows. We will introduce some candidate cyclic groups where we believe the DDH problem, CDH problem and the discrete-log problems are indeed difficult to solve. Namely, we will introduce the prime order cyclic groups.

(Refer Slide Time: 00:46)



So let us begin our discussion with the importance of good cyclic groups. So remember that in the last lecture we have seen the definition of the DLog assumption, CDH assumption and DDH assumption. So if we are designing any cryptographic primitive whose security is based on the hardness of these problems, then we need to appropriately choose the underlying cyclic groups.

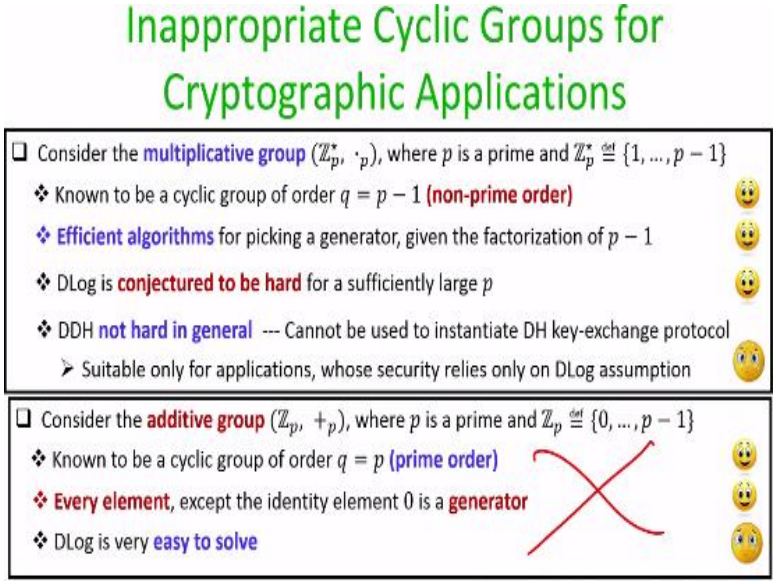
Because if the underlying cyclic groups that we use to instantiate the cryptographic primitive, if in those groups these problems are easier to solve, then the resultant cryptographic primitive will no longer be secure. So now the interesting question is for which cyclic groups or for which candidate cyclic groups indeed these problems, namely the DLog problem, the CDH problem and the DDH problem are difficult to solve.

So remember in the last lecture the concrete steps of the Diffie–Hellman Key Exchange Protocol were given assuming that we are operating on a group where the DLog, CDH and DDH problem are indeed difficult. But now our question is how exactly we find those groups. So there are 2 popular choices of cyclic groups or candidate cyclic groups, where we believe that these problems are indeed difficult to solve.

The first choice is the groups of prime order, namely groups where the number of elements is some prime number. However, it turns out that not all the prime-order cyclic groups are appropriate for cryptographic applications. In fact, we will see some candidate prime-order cyclic groups where DLog problem, CDH problem, DDH problem are indeed very easy to solve.

But it turns out that we have other types of prime-order cyclic groups, where we strongly believe that DLog, CDH, DDH problem are indeed difficult to solve. The second choice of picking the candidate cyclic groups, is the group based on points on elliptic curves. So in this lecture we will consider the groups of prime-order based on certain properties. In the next lecture we will see the cyclic groups based on the points on elliptic curves and then we will compare these 2 types of groups, which one is better to instantiate the cryptographic primitives, whose security is based on the hardness of DLog, CDH and DDH problems.

(Refer Slide Time: 03:07)



So before we proceed further, let us see the inappropriate cyclic groups for cryptographic applications, namely which cyclic groups we should avoid for instantiating cryptographic primitives, whose security is based on DLog, CDH and DDH assumption. So we start with the multiplicative group, namely the set $\mathbb{Z}_p^*$, where $\mathbb{Z}_p^*$ consists of the elements 1 to $p-1$ and my underlying operation is multiplication modulo p.

So remember multiplication modulo p is defined as follows. If you want to perform the multiplication modulo p of 2 numbers a and b from the set $\mathbb{Z}_p^*$, then you perform the integer multiplication ab and take the remainder, by doing a mod p operation, which ensures that the resultant remainder is in the set 0 to $p-1$. It turns out that this set $\mathbb{Z}_p^*$, along with this multiplication modulo p operation, constitutes a cyclic group of order $p-1$.

Why of order $p-1$? Because the elements in this set $\mathbb{Z}_p^*$ are the elements 1 to $p-1$, so it has $p-1$ number of elements. And since p is prime, $p-1$ cannot be prime, that is why the order of this group is a non-prime number. And we can prove that this group is a cyclic group. And we have efficient algorithms for picking a generator for this group given the factorization of $p-1$. So remember the Diffie–Hellman key-exchange protocol steps that we had seen in the last lecture, the public set up that we need there is the description of the group, where as part of the description of the group, the details of the generator should also be publicly known. So we need polytime algorithm for picking the generator and when I say polytime algorithm, I mean polynomial in the number of bits that we need to represent the element of the set $\mathbb{Z}_p^*$. So

we have polytime algorithms, efficient algorithm for picking generators for this group, given that you are given the factorization of $p-1$.

So that is also a positive of this group. Also it is believed that the DLog problem is indeed difficult to solve in this group, provided p is sufficiently large. So that is also a good news with respect to this group. But the problem here is that DDH problem is not hard in general in this group and that means we cannot use this group to instantiate the Diffie–Hellman Key Exchange Protocol. Because remember for the security of the Diffie–Hellman Key-Exchange protocol, namely for the strong-privacy of the Diffie–Hellman Key-Exchange protocol, we need that the DDH problem should be difficult to solve in the underlying group. So if I use $\mathbb{Z}_p^\star$, along with the operation multiplication modulo p as my underlying cyclic group and perform operations as per the steps of the Diffie–Hellman Key-Exchange protocol, then it is not guaranteed that the resultant key-exchange protocol satisfies the notion of strong-privacy. Because it is not guarantee that the DDH problem in this specific group is hard. So that means this group is suitable only to instantiate those applications or those cryptographic primitives, whose security is just based on the DLog assumption and not on the DDH assumption, which is not the case for the Diffie–Hellman Key-exchange protocol.

So that is the bad news with respect to this group. So that means you can see now the group $\mathbb{Z}_p^\star$ with operation multiplication modulo p may not be suitable group to instantiate the Diffie–Hellman Key-exchange protocol, it is an inappropriate cyclic group.

Now consider the additive group. So the previous group $\mathbb{Z}_p^\star$ with the operation multiplication modulo p was a multiplicative group. Now consider an additive group, where my set is $\mathbb{Z}_p$, consisting of the elements 0 to $p-1$ and my operation is addition modulo p, where addition modulo p on elements a and b is defined as follows: you perform the integer addition of a and b and take the reminder with respect to the modulo p. That is the way we define addition of a and b modulo p. So with respect to this group the following facts are known. First of all, this group is known to be a cyclic group and that too of prime order. Because the elements in the set $\mathbb{Z}_p$ are 0 to $p-1$, namely it has p number of elements, where p is a prime and it is known to be a cyclic group. So that is a good news. And another interesting property of this prime-order group, is that every element, except the identity element is a generator and this is not only specific to this group. This property holds with respect to any group which has a prime order.

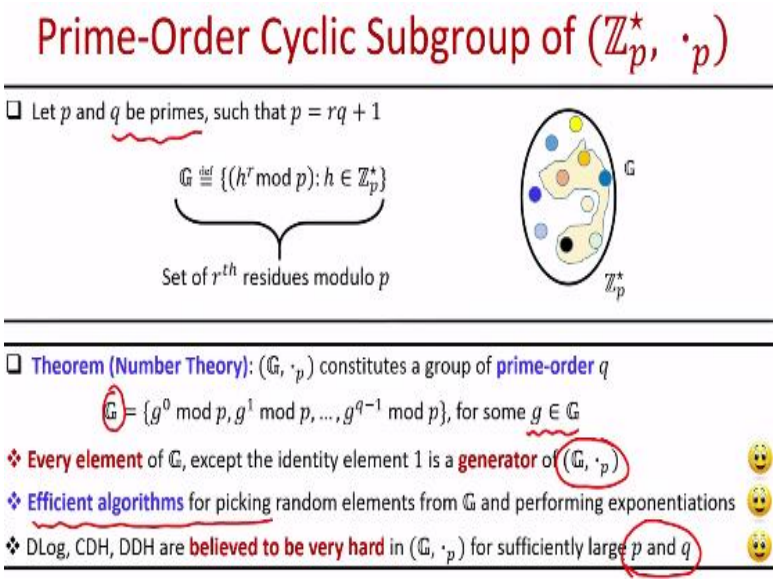
The fundamental fact which comes from abstract algebra is that if you have a group whose order is prime, namely it consists of prime number of elements, then any element from that group except the identity element of that group is a generator. So picking generator is not at all going to be a sophisticated task. If we operate or if we perform operations in this group, you can pick any element except 0, that is bound to be a generator.

However, the most unfortunate part or the most unfortunate fact with respect to this group is that the DLog problem is very, very easy to solve in this group. In poly amount of time you can compute the DLog of any randomly chosen element from this set with probability 1. So what exactly will be an instance of the DLog problem in this group? So you pick a random index α in the set 0 to $p-1$, because your group is of size p. And you compute αg, where g is the publicly known generator. And say the resultant output is u and the challenge for you is to compute this unknown α, such that αg modulo p would have given you u. So what are the things known to you, you are knowing g here, you are knowing u here and you know that everything is related modulo p here and your goal is to find out α here.

It turns out that we can easily compute α here, by multiplying both the side with the multiplicative inverse of g. And multiplicative inverse of g modulo p can be computed in polynomial amount of time. And if you multiply both the sides with multiplicative inverse of g, then the effect of g cancels out, and what you are left with, is α. And hence you are obtaining the value of α, namely the discrete log of u to the base g in polynomial amount of time.

So I am not giving the full details of the discrete log solver for this group, but that is the overall idea. So even though this specific group has some nice properties, namely it is a prime order cyclic group, picking generator is not a difficult task, the most unfortunate part here is that the DLog problem is very, very easy to solve and that automatically implies that the CDH problem is easy to solve. And which automatically implies that DDH problem is also easier to solve in this group. So that means this group cannot be used at all to instantiate any cryptographic primitive, whose security is based on the hardness of DLog problem, CDH problem, DDH problem.

(Refer Slide Time: 10:24)



So now let us see another group, which is quite appropriate to instantiate cryptographic primitives based, on the difficulty of DLog and CDH and DDH problems. And this group is basically a prime-order cyclic sub group of the group $\mathbb{Z}_p^\star$, with the operation multiplication modulo p. So let p and q be primes, publicly known, such that p is related to q in this form, namely $p = rq + 1$, where r is also publicly known.

And then let me define a set G to be the set of all element of the form h^r modulo p, where h belongs to the set $\mathbb{Z}_p^\star$. So pictorially what I am doing here is, you imagine the elements in this bigger circle as the elements of $\mathbb{Z}_p^\star$ and $\mathbb{Z}_p^\star$ has $p - 1$ number of elements, because my underlying operation is multiplication modulo p. What I am doing here is in the set G, I am just collecting a smaller subset of elements from $\mathbb{Z}_p^\star$.

Namely I am collecting all the r^{th} residues modulo p. Why r^{th} residues? Because I am taking h from $\mathbb{Z}_p^\star$ and raising it to the power r and taking modulo p. And that is why the result can be viewed as r^{th} residue modulo p. For instance, if $r = 2$, then basically G consists of all the elements, which are perfect squares modulo p, because I will be collecting elements of the form h^2 mod p, where h belongs to $\mathbb{Z}_p^\star$.

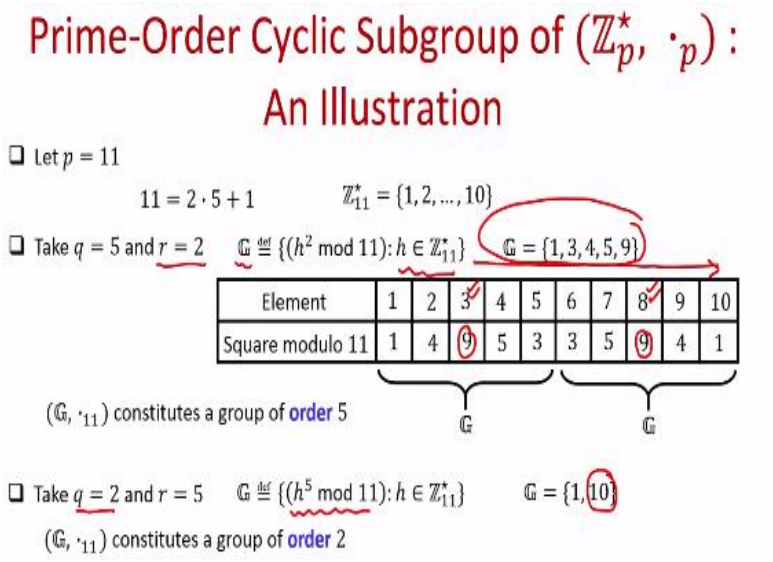
Whereas if r would have been 3, then G basically consists of all the elements which are perfect cube modulo p and so on. So in general if G is some h^r, then G is the set of all r^{th} residues modulo p. So that is the way I am computing this set G here. And a very interesting result from number theory, which I am not going to prove here, explicitly states that the collection G or the

subset G, which is a subset of $\mathbb{Z}_p^\star$, along with the operation multiplication modulo p, constitutes a group of order q.

Namely the set G will have q number of elements. And since I have selected q to be a prime, that means the order of G is a prime. And we can prove that if you perform the operation multiplication modulo p on the elements that we have selected in the set G, then it satisfies the group axioms. Namely we can prove that the elements in G can be expressed as the powers of a generator, where the indices of the powers will be in the range 0 to $q-1$, for some generator g, belonging to the set G. And moreover the important interesting property that we obtain here is that since my set G or group G has prime order, every element in this set G, except the identity element, will be a generator for the subgroup which I have picked. Why I am calling it subgroup? Because the elements in the set G is a subset of the elements in $\mathbb{Z}_p^\star$. But the operation in this set G, namely multiplication modulo p, is the same as the operation in the bigger group namely $\mathbb{Z}_p^\star$. That is why I am calling it as a subgroup. So since the order of this subgroup is prime, every element from this subgroup will be a generator, except the identity element. Moreover, for the subgroup that I have chosen here, we have efficient algorithms for picking random elements as well as for performing group exponentiation.

So if you recall the steps of the Diffie–Hellman key-exchange protocol, their sender and receiver have to pick random elements from the underlying group over which they are performing the operations. So for that we need to have efficient algorithm, polytime algorithms, for picking random elements from the group. And it turns out that if I set my group to be the set of all r^{th} residues modulo p, then I have an efficient algorithm for picking random elements from this group and for performing group exponentiation. And it turns out that DLog, CDH, DDH, all these problems are believed to very, very hard for sufficiently large values of p and q, if I pick my subgroup G to be the set of all r^{th} residues modulo p.

(Refer Slide Time: 14:58)



So let us see an illustration of prime-order cyclic subgroup of $\mathbb{Z}_p^\star$. So imagine $p = 11$, where $p = 11$ is a prime number. So I can express 11 in the form $2 \times 5 + 1$ and $\mathbb{Z}_{11}^\star$ basically consists of the elements 1 to 10. So if I take my q to be 5, then p and q are related as 2 times 5 + 1. So I can take my r to be 2 and what I can do is, I can set my set G to be all the perfect square modulo 11.

Namely I take all the elements h from the set $\mathbb{Z}_{11}^\star$ raise it to the power 2 and do modulo 11. And the resultant output is my collection G. So what I have done in this table, is I have taken all the elements from the set $\mathbb{Z}_{11}^\star$ and the resultant squares and if I take the resultant squares I obtain my set G. Namely my set G consists of the elements 1,3, 4, 5, 9 and you can see in the table, that under the squares, the elements 1,3, 4, 5, 9 are repeated twice.

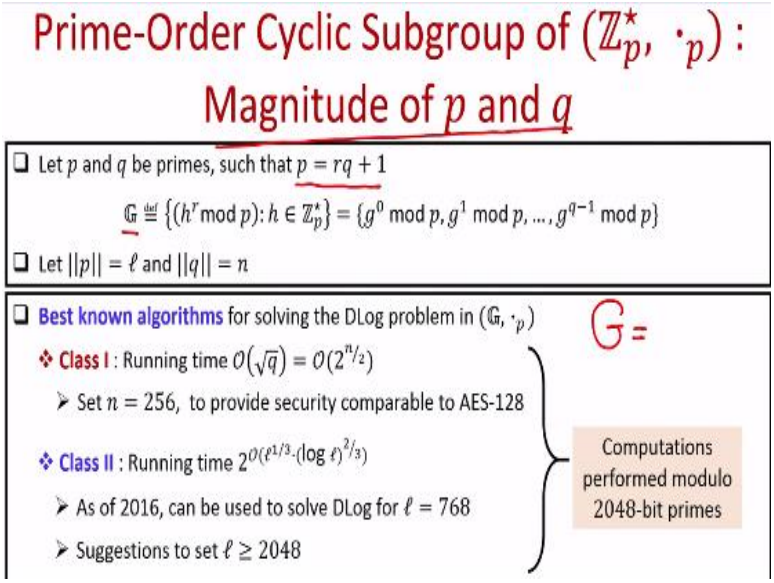
So 1^2 gives me 1, and so does 10^2. 2^2 gives me 4, so does 9^2. 3^2 gives me 9 and 8^2 gives me 9 and so on. This is because any perfect square, namely an element in the set G will have 2 square roots, because it is a square residue. So it will have 2 square root modulo p. And if one of the square roots is a then the other square root will be $-a$. And $-a$ here is nothing but $p - a$.

So for instance if I take the element 9, which is an element of G, then it has 2 square roots because it is the result of 3^2, so that is why 3 is one of the square roots. And similarly 9 is the result of 8^2 modulo 11 and that is why 8 is also one of the square roots of 9. And it is easy to

see that the subset 1, 3, 4, 5, 9 along with the operation multiplication modulo 11 constitutes a group of order 5.

You can verify that in the same way, for the same example where $p = 11$, I can take my q to be 2 and accordingly r will be 5. And now if I focus on the fifth residues modulo 11, namely the collection of all h^5 module 11, where h belongs to $\mathbb{Z}_{11}^{\star}$, then I obtain the subset 1, 10 and this subset 1,10, we can see that it actually constitutes a cyclic group of order 2. Because it has 2 elements and it has the identity element and 10 is the generator. So that is an illustration here. But I have not proved the generic results. Namely if I take p and q to be on the form $p = r \times q + 1$, then the set of all r^{th} residues gives you a cyclic group. I am not proving that, you can see any of the standard references for number theory for the proof of that fact.

(Refer Slide Time: 18:11)



So now the question is, we have seen that we can form a prime-order cyclic subgroup of $\mathbb{Z}_p^{\star}$ and DLog, CDH, DDH problems are believed to be very difficult in those cyclic subgroups. The questions turns out that what should be the magnitude of the resultant p and q to ensure that indeed the DLog, CDH and DDH problems are difficult in the resultant subgroups.

So the problem that we want to address here is we are given p and q which are primes, where p is of the form $rq + 1$. And we have found a set of r^{th} residues, which we know is a cyclic group, which has a generator g and say the number of bits that we need to represent p is ℓ and

the number of bits that we need to represent q is n. Now the best known algorithms for solving the discrete-log problem in the subgroup that we have formed here, it falls under 2 categories.

We have the Class I algorithm that we know for solving the discrete log problem. Their running time is of order $\sqrt{q}$. And now since q is of the magnitude 2^n, that means $\sqrt{q}$ will be of the magnitude $2^{\frac{n}{2}}$. So even though this is exponential time in the underlying security parameter, we have to very judiciously decide the value of n, when we are instantiating this subgroup for instantiating a cryptographic primitive whose security is based on the DLog assumption.

It turns out that if you set $n = 256$, namely if we select q which is a 256 bit prime and accordingly set a prime p where p and q are related by the relationship that p is $rq + 1$, then just by setting $n = 256$, we achieve a level of security which is comparable to AES-128. So remember when we saw the practical instantiation of block cipher, like AES, DES, where we aim for the practical security and by practical security of AES-128, I mean that the best possible attack that an adversary can launch to recover an AES key, where the adversary is given several (x, y) pairs, where x is the AES input and y is the corresponding AES output, under an unknown key, then the complexity of the best known attack should be of order 2^{128}. So it turns out that if we instantiate any cryptographic primitive based on the hardness of DLog problem by selecting a cyclic subgroup of $\mathbb{Z}_p^\star$ by setting $n = 256$, then the best known algorithm for solving the DLog problem by this class of algorithm will take time roughly of order $2^{\frac{256}{2}}$, which will be of order 2^{128}. That means we get the same level of security, as your AES-128 would have provided.

Whereas the Class 2 algorithms for solving the DLog problem in this cyclic subgroup, its running time is of order 2 to the power order poly logarithmic in the number of bits that we need to represent p. So as of 2016, this Class 2 algorithm can be used to solve instances of DLog problem and for any instantiation of the cyclic subgroup where ℓ is 768. And it is suggested that to have meaningful notion of security we should operate by setting ℓ to be 2048.

That means if we summarize, to tackle the class 1 algorithms and Class 2 algorithms that we have for solving the DLog problems in this cyclic subgroup, to ensure that we have reasonable amount of security or the running time of the adversary for solving the discrete log problem is of sufficiently large order, we have to perform computations modulo 2048 bit prime number.

That means say for instance if we use the Diffie–Hellman Key-exchange protocol and if we instantiate the steps of the Diffie–Hellman Key-exchange protocol by setting my set G to be set of all r^{th} residues modulo p, then I have to ensure that my p should be a 2048 bit large prime number. That means both sender and the receiver have to perform computations modulo this large prime number, which actually reduces the running time of both sender and the receiver.

So even though we have now a candidate cyclic subgroup which we can now use to instantiate any cryptographic primitive, whose security is based on the hardness of DLog, CDH and DDH assumptions, it turns out that the running time of the sender, receiver or all the involved parties also reduces. Because we are going to perform operations modulo a very large prime number.

So an interesting question will be, can we have other kind of candidate cyclic groups, where we do not have to perform operations modulo such a huge prime number, but still the CDH problem, DDH problem and DLog problems are difficult to solve in those alternative groups. Snd in the next lecture, we will see one such candidate group. So that brings me to the end of this lecture.

In this lecture we have introduced one candidate cyclic group, namely the cyclic subgroup of $\mathbb{Z}_p^\star$ with respect to the operation multiplication modulo p, and we believe that the CDH problem, the DLog problem and DDH problem are indeed difficult to solve in this group. However to obtain practical level of security, we have to set the value of the modulus p to be a very large number, to ensure that sender and receiver obtain reasonable amount of security, which actually ends up making the running time of the sender and the receiver also slow. Thank you.