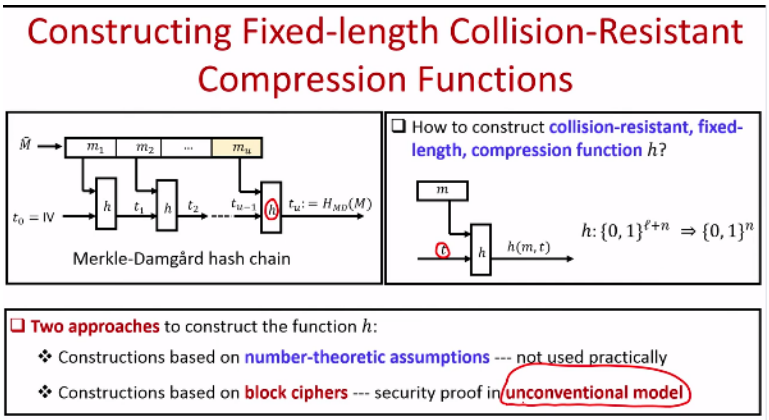
Foundations of Cryptography
Prof. Dr. Ashish Choudhury
(Former) Infosys Foundation Career Development Chair Professor
Indian Institute of Technology – Bangalore

Lecture – 28
Cryptographic Hash Functions - Part II

(Refer Slide Time: 00:32)

Hello everyone, welcome to this lecture. So just to recall, in the last lecture we had seen the Merkle-Damgard paradigm for constructing collision-resistant hash functions for any size inputs and in this lecture we will see the stage 1 of the Merkle-Damgard paradigm, namely how to construct a fixed-length compression functions.

(Refer Slide Time: 00:49)



So just to recall pictorially, this is how a Merkle-Damgard paradigm will take any fixed-length compression function little h which is collision-resistant and apply it iteratively to obtain or hash function which can hash inputs of any size. So now the interesting question is

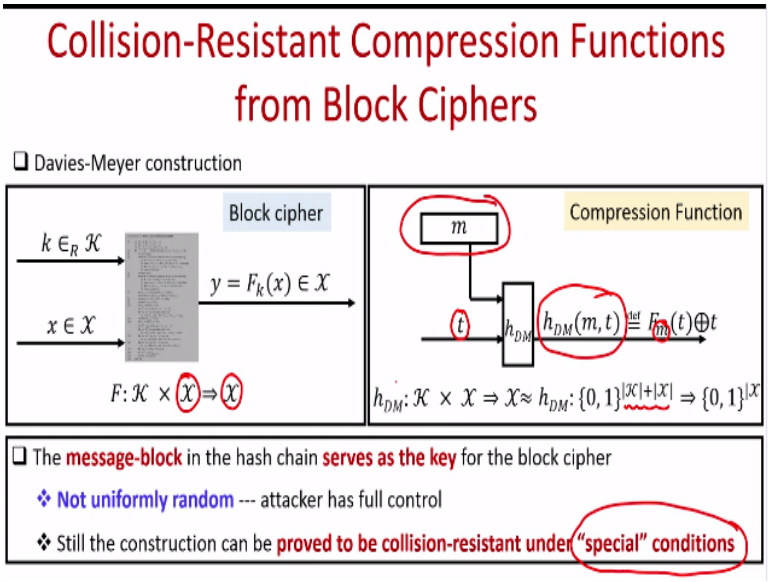
how exactly you obtain this fixed-length collision-resistant compression function little h at the first place. So pictorially, this function little h takes an input of size l + n bits which can be passed as 2 inputs and input m of little l bits and another input t of little n bits.

Together it will compress it and gives you an output of size little n bits right. So that is how you can interpret this fixed-length compression function and it turns out that there are 2 approaches to construct this function h. The first approach is to make a design construction based on number-theoretic hard problem or number-theoretic assumptions. So when we will start our discussion on number theory and public key cryptography, we will come back and we will see how to use some number-theoretic hardness assumptions to construct this function little h.

This approach is not used practically because even though the constructions, their running time, they are polynomial in the security parameter, the actual running time when deployed in practice is of order of several magnitudes and that is why we do not use the instantiations of this function little h based on number-theoretic hardness assumptions. Instead what we do is, we design constructions, we go for constructions based on block ciphers or we can take any of the existing block ciphers say aes, tes or we can design dedicated block ciphers.

These are the construction or so these are the instantiations of the function little h, which we use in practice. However interestingly the security proof of the constructions based on this block ciphers, they are in very unconventional model. They are unconventional model in the sense, we make very strong assumptions from the underlying model and then give the security proof.

(Refer Slide Time: 03:06)



So let us see this approach of how to construct collision-resistant compression function, fixed-length compression function from block ciphers and there are several constructions available. We will see one of the constructions which is called as Davies-Meyer construction. So what you are given is a block cipher which you can interpret as a keyed pseudorandom permutation, taking a key from the key space and input x from the block space and giving you an output, where the output and the block size are the same.

Namely both of them belong to the set this fancy x and using this we design a compression function little h which I denote as h sub DM, namely we can call it as Davies-Meyer compression function, which takes an input right of size this cardinality of k and cardinality of x space, so namely it takes 2 inputs, 1 input is considered interpreted as m and other input is interpreted as t and the output of this compression function is defined as follows.

We evaluate the block cipher with respect to the key little m and treating the t part as the block and the output is again XORed with the t part of the input of this Davies-Meyer hash compression function, so that is how this Davies-Meyer compression function is evaluated, right. So the interesting part here is that the message block in the hash chain, so remember that this function little h is going to be plugged in in the hash chain of the Merkle-Damgard paradigm right when we are evaluating the overall hash function big H, right.

That is why pictorially I am representing this function little h consisting of 2 inputs, one input will be the block part of the message which we want to hash in the bigger message and one will be the t part which will be coming from the outcome of the previous invocation of the

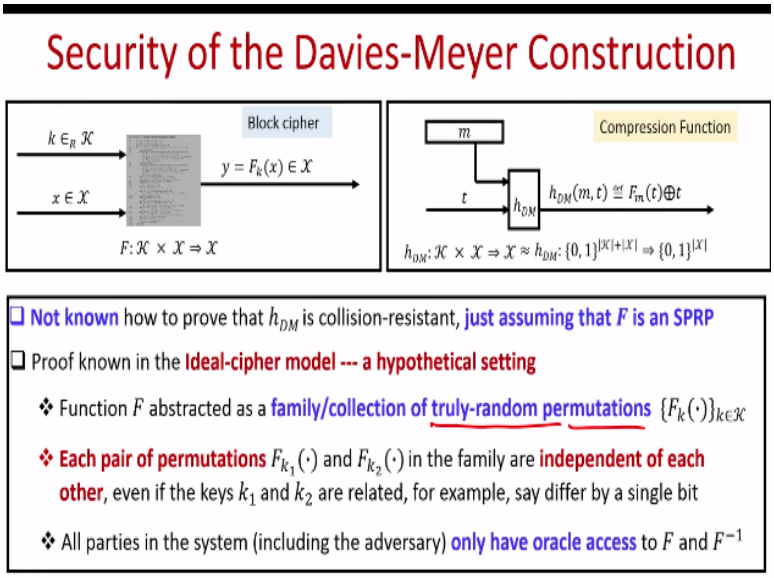
little h function right and the way we are designing this Davies-Meyer compression function is basically the message block on which we are going to operate this Davies-Meyer instruction will be treated as the key for the pseudorandom permutation of the block cipher. It need not be uniformly random right.

Because attacker could have full control over the bigger message which it want to hash as per the Merkle-Damgard paradigm and that means it gives the attacker full control over what exactly is serving as the key for the block cipher when the block cipher is internally used in the design of the Davies-Meyer compression function. However even though the adversary has the full control over the key which is used in the underlying instantiation of block cipher in this Davies-Meyer compression function, it can be proved that the overall construction of this Davies-Meyer compression function is indeed collision-resistant under special conditions, right.

That means there is nothing to worry about here, even though you might be wondering that or we have discussed that block ciphers remain secure only if the key is not known to the attacker and the key is uniformly random, but the way we are operating or using the block cipher in this Davies-Meyer compression function is that, the block of the message which we are going to use or apply the hash function little h right, it is fully under the control of the adversary and that is actually serving as the key for your blocks underlying instantiation of your block cipher and that need not be random.

But still we can prove that under special conditions the way this output of the Davies-Meyer construction is defined overall the function, Davies-Meyer construction that we have constructed here is indeed collision-resistant, right.

(Refer Slide Time: 07:05)



So here is how we have constructed the Davies-Meyer compression function and as I am continuously saying that we can prove the collision resistance of the Davies-Meyer construction under special assumptions, this is because we do not know how to prove the collision resistance of the Davies-Meyer construction that we have given just assuming that the underlying block cipher is a strong pseudorandom permutation. It is not sufficient to just assume that the underlying function F is a strong pseudorandom permutation and give a proof that the Davies-Meyer construction is indeed collision-resistant.

However, interestingly the proof of the collision resistance of the Davies-Meyer construction is known in the ideal-cipher model which is kind of a hypothetical setting, because it makes very strong assumptions about your underlying block cipher. So let us see what exactly is this ideal-cipher model? So in ideal-cipher model, the underlying block cipher or the keyed permutation F is abstracted as a collection of several truly random permutations, where each member of this family or each member of this collection is indexed by the underlying key with which we are going to operate the function F.

So remember, the block cipher it takes a block input and a key input. So what basically we are assuming in this ideal-cipher model is that even though we are given the description of a single F function as soon as we fix the value of k that gives an instantiation of the F function, which can be treated as one member of a bigger family. So different values of the k which are used in the instantiation of your function F will give you different members of the bigger family and that is why we are now viewing that even though we are given the description of a single F function when used with different K.

It gives you different members from a bigger family and that is what we mean by function F abstracted as a collection of truly random permutations. The second property in this ideal-cipher model is that once we know a family member Fk1 and another family member Fk2, namely these are the instantiation of your block cipher with the key k1 and the key k2 the assumption here is that the family member Fk1 and the family member Fk2 namely the keyed permutations Fk1 and Fk2**,** they are independent of each other even if the keys k1 and k2 are dependent or related to each other.

That means even if it so happens that k1 is almost the same as k2 except say the last bit, the assumption in this ideal-cipher model is that the function Fk1 will be completely independent in the sense its output will be uniformly random and it will be completely independent from the outputs of the function Fk right, so that is a second property in this ideal-cipher model. So till now there is nothing unconventional here that we are assuming about the function F.

Because when we are assuming that function F is a strong pseudorandom permutation that means even though it is not a truly random permutation, it gives you pseudo randomness guarantee that means we can assume that it behaves like a pseudorandom permutation, but in the ideal-cipher model we are making slightly more stronger assumptions, we are assuming that each member of this bigger family is indeed a truly random permutation and independent of each other.

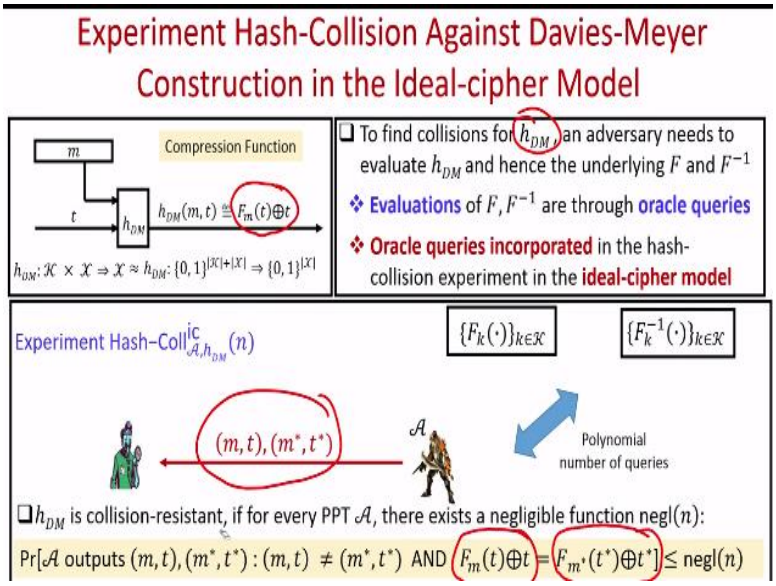
The strongest property or you can call it as the most unconventional property in this ideal-cipher model is that no party or no entity in the system who is going to use a cryptographic primitive, which makes use this block cipher will have access to the code of the function F and the code of the function F inverse. That means if at all any entity wants to compute the value of the function F with respect to some key, it will make oracle access or it will make oracle calls to the keyed permutation Fk and it will just get back the value of the keyed permutation on the input for which it wants the value of the Fk.

In the same way if some entity wants to know the value of F inverse with respect to some input key k and some input y, then it can make just oracle call to the F inverse oracle and it will get back the corresponding output, but it will not get any access to the code of the block cipher. So that is the most unconventional assumption that we are making in the ideal-cipher

model because in reality when we are designing the Davies-Meyer construction and instantiating it we need the actual block cipher which we are going to plug-in in this construction of the Davies-Meyer construction.

Every entity who is going to use that instantiation of the Davies-Meyer construction will know the code of the function F including the adversary, but when we are analyzing the security of the Davies-Meyer construction, the assumption that we are making in the ideal-cipher model is that no one will be knowing the code of the function F, it is just a collection of several random members, if you want to talk to one random member, just make an oracle call right.

(Refer Slide Time: 12:17)



So assume now we are in the ideal-cipher model, what we are going to prove is that this Davies-Meyer construction of fixed-length compression function that we have given here indeed is a collision-resistant function. So remember that indeed it is a compression function because we are taking an input of size n + l bits and giving you an output of size l bit, so it is a compression function.

What we have to prove is that in polynomial amount of time, it is difficult for an adversary to come up with a pair m, t and m star, t star such that the output of the Davies-Meyer function is the same where the Davies-Meyer construction is using a block cipher right. So since we are going to analyze the security of this compression function in the ideal-cipher model, any adversary who would like to find out a collision for this Davies-Meyer construction it needs

to evaluate the Davies-Meyer function on several messages of its choice right, then only it could come up with a collision.

To evaluate the Davies-Meyer function on messages of its choice, internally it needs to evaluate the function F and F inverse as well because we do not know what exactly is the strategy of the adversary; it may ask, it may evaluate the underlying F on several m comma t pairs, it could evaluate the underlying F inverse also on several messages of its choice or several inputs of its choice, and then only it could come up with a corresponding collision.

But since we are now in the ideal-cipher model what we are going to assume is that even the adversary who would like to evaluate the underlying F and F inverse on inputs of his choice to come up with a collision for this Davies-Meyer function, we will make oracle queries right, because it would not be knowing the code of the function F and code of the function F inverse. So that is why to make this oracle queries to incorporate this oracle queries from the adversary to find out the collision.

This oracle queries are incorporated in the hash collision experiment when we analyze the security of this Davies-Meyer construction in the ideal-cipher model. So remember in the actual hash collision experiment in the non-ideal-cipher model or in the standard model, adversary can query anything, adversary did not have to query anything, because it will be having access to the code of the function F or F inverse if we are using any F or F inverse in the design of the underlying hash function.

But now since we are in the ideal-cipher model, even adversary's interaction with the function F and F inverse will be through oracle queries and that is why they also need to be incorporated in the modified hash collision experiment. So the modified hash collision experiment is as follows. So we now assume that both the experiment as well as the adversary will have oracle access to the family of the functions Fk and the family of the functions Fk inverse.

So you can imagine this oracles Fk and Fk inverse are kind of lying in the sky where no one can see what exactly is the code of the function Fk and Fk inverse and if anyone wants to compute the value of the function Fk for k of its choice, it just shouts to the sky that please guy give me the value of this function Fk on this input and it will see the value of Fk on that

input and same way the interaction with Fk inverse is handled here right. So that is why I am putting this family of functions Fk and Fk inverse inside a box here.

No one can see what exactly is happening inside the box, all the interactions are going to be just an interface, you supply some input and you get back the output. So the modified hash-collision experiment is as follows. So remember the goal of this adversary is it knows the description of your Davies-Meyer hash function, namely it knows that the Davies-Meyer's hash function is basically F of t under the key m XOR t with the t part right.

The goal of the adversary is basically to come up with a pair m, t and m star, t star which constitutes a collision with respect to this Davies-Meyer function, but since we are in the ideal-cipher model any value of F which adversary would like to compute or any value of F inverse which the adversary would like to compute will be in all made through oracle queries. So imagine if adversary would like to know the value of F with respect to this key ki and on the input xi.

So, it can make an oracle query and what the oracle is going to return back is it is going to return back the value of the function F with respect to the key ki on the input xi, and if that adversary sees this response, then basically what adversary sees is the value of the Davies-Meyer function on the input pair ki, xi because the value of the Davies-Meyer function on this combined input ki, xi, will be the value of the function or the block cipher F on the input xi with the key ki XOR with the input xi.

But since adversary is then the ideal-cipher model, he cannot compute the value of F of xi with the key ki that is why it has made an oracle query. So that is how the adversary is going to compute the value of Davies-Meyer functions on messages of its choice. It turns out that this is not the only way by which the adversary could compute the value of the Davies-Meyer function on inputs of his choice.

It could take the help of the inverse oracle access as well, say for instance it can ask hey inverse oracle give me the value of the inverse of the function under the key ki on the input yi and in response what the oracle is going to do is it is going to return back the inverse of this function on the input yi under the key k, sorry under the key ki, sorry for the typo error, it should be under the key ki right, and once the adversary learns the inverse of this yi under the

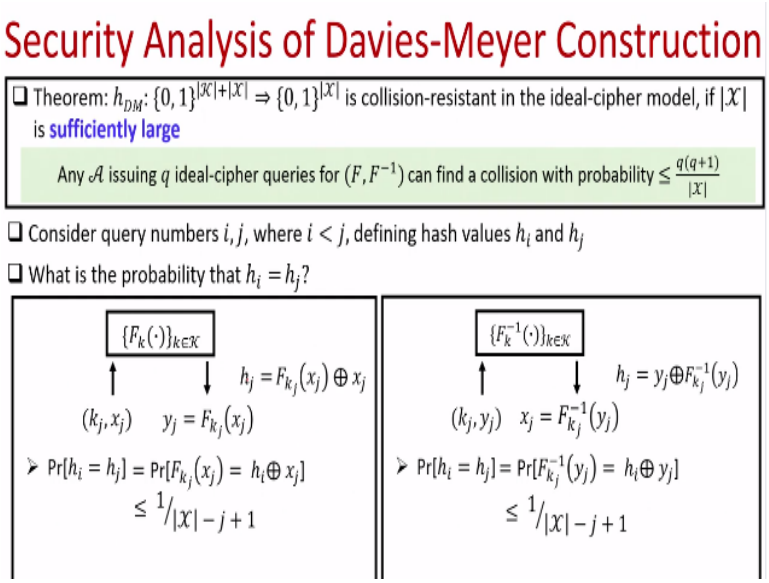
key ki that gives him the value of the Davies-Meyer function on the input ki, xi because the way Davies-Meyer function is constructed.

The output of the Davies-Meyer function on this input ki, xi would be like this. So by making oracle queries to the Fk family and by making oracle queries to the Fk inverse family, we can assume that the adversary is now going to compute the output of the Davies-Meyer function on several messages of its choice, namely polynomial number of messages because our adversary is polynomially bounded and now finally he submits a collision, namely a pair of inputs m, t and a pair of input m star, t star.

The security definition here is that we will say that this collision-resistant experiment is successful for the adversary in the ideal-cipher model if in this model where adversary is now making or given oracle access to the Fk family and Fk inverse family, the probability that the adversary could come up with a collision is upper bounded by a negligible function and the collision is now specific because now we are analyzing the collision with respect to the Davies-Meyer construction.

So, the output of the Davies-Meyer construction for the m, t message which the adversary is submitted will be this and the output of the Davies-Meyer function for the message m star, t star that the adversary has submitted will be this. Our goal is to analyze what is the probability that this condition hold. The security definition is we will say that the Davies-Meyer construction is collision-resistant in the ideal-cipher model if the probability that any poly-time adversary could come up with a special m, t and m star, t star satisfying this condition is upper bounded by some negligible function.

(Refer Slide Time: 20:15)



What we are going to prove is that indeed the Davies-Meyer construction that we have given constitutes a collision-resistant function in the ideal-cipher model provided the size of the x domain or the x set is sufficiently large. More specifically, what we are going to prove here is that any poly-time adversary or any adversary A issuing q number of ideal queries altogether to the F family and F inverse family can find a collision with probability upper bounded by q times q+1 over the size of the x set, right.

So here q is the total number of queries which the adversary is allowed to make to the F family and F inverse family right. So it is not q for the F and q for F inverse altogether, right. So if he is making s number of queries to F and t number of queries to F inverse, then s + t is basically q and what we are going to prove is that in the ideal-cipher model any poly-time adversary making q number of queries, his success probability of finding a collision and the Davies-Meyer function is upper bounded by this quantity.

So what we are going to do here is we consider the query numbers i, j, where i is less than j that means assume that adversary has already queried up to j-1 queries altogether to the combined F and F inverse family and now he is making the jth query and any time when it is making the query the response from the queried gives him the value of the Davies-Meyer function on some input pair right, because that is we had already seen that in the last slide.

So if he is making queries to the F family, then based on the response it gets the value of the Davies-Meyer function on some input or if he is making a query to the F inverse family, then again based on the response which gives him the value of the Davies-Meyer function on

some input. So what we denote here is that the ith query based on the response whatever hash value he learns I denote it as h sub i and based on the response for this jth query whatever hash value that he learns for the Davies-Meyer function I denote it as h of j.

Now I am interested to compute what is the probability that the ith hash value and the jth hash value which the adversary learns in this ideal-cipher model are the same. So now there could be 2 cases here. Assume that, the jth query which the adversary is making is for the F family. So when he is making a jth query to the F family, then he can make a query of this form that means basically he is asking for the value of the family member F of kj on the input xj.

So he will see the response which I denote as yj and based on the response he basically learns the value of the Davies-Meyer function on some input, basically he learns the value of the Davies-Meyer function on the input kj, xj and that is denoted by as h of j and this will h of j will be this value right. So we are interested to analyze what is the probability that the jth hash value which the adversary has learnt by making this query to the Fk family it is the same as the ith hash value which the adversary has already learned while making the ith query and remember the ith query could be either to the Fk family or to the Fk inverse family.

So the probability that the ith hash value which the adversary has learnt is the same as the jth hash value which the adversary has just now learnt are same, it is the same as the probability that the family member Fk of j when evaluated at the input xj gives you this output. If that is the case then h of i and h of j will be the same, but since we are in the ideal-cipher model right, what is the value of the family member Fkj on the input xj, well it is a uniformly random value independent of each other right.

It is a uniformly random value over the x set and it it is different from all the previous Fkj values for which the adversary might have already asked the output value right. So, remember we are in the case where i is less than j, we are analyzing the probability with respect to the collision for the output of ith query and jth query. So in the worst case it may so happen that for all the previous, j – 1 queries the adversary might have asked for the value of the function family with respect to the key kj.

In response the adversary might have seen the output of the family member Fkj for various inputs and since Fkj is a permutation right, when the oracle is giving him back the response for the Fjk family on the jth input that has to be different from all the previous j – 1 output that oracle has already returned to the adversary. So the probability with which this condition namely the output of the jth query under the key Fkj is equal to this is always upper bounded by 1 over the size of the x set - J + 1.

So that is a probability with which, by making the jth query to the Fk function, the adversary can hope that it gets a collision with respect to the output of the ith query, which he had made in the previous interaction with the oracle. On the other hand, it may so happen that the adversary's jth query is for the F inverse function right. So it could ask for the value of the inverse, it could ask for the oracle service for this input, from the F inverse oracle and in response it gets back the output, basically it sees the inverse of the input yj under the key Fkj right.

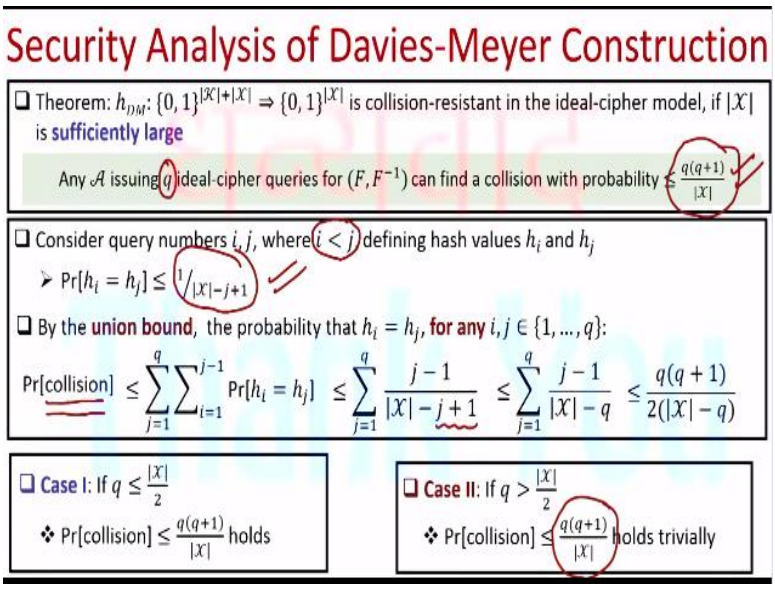
So he is basically asking for the access to the F inverse kj member on the input yj and it gets back the output xj and based on that it learns the jth hash value right on this message, namely the value of the Davies-Meyer function and that the output h of j will be this. Now in this case, what is the probability that the jth value which the adversary has learnt here is the same as the output of the ith query which the adversary has already learnt in the previous interaction?

Well, the probability of that is the same as that the probability that your F inverse function under the key kj on the input yj gives you an output which is same as XOR of hi and yj and again we can run the same argument that we have done for the case on your left-hand side right. Remember that in the ideal-cipher model, each of the family members Fk and Fk inverse, they are you truly random and independent of each other.

So that means this family member F inverse kj is like a truly random permutation and it may so happen that all the previous j - 1 queries, the adversary could have made with respect to the family member kj, that means in the worst case he might have asked for the value of F inverse kj on various y values of his choice for all the j - 1 queries. So now for the jth query when the oracle is giving him back the response, the response is going to be different from all the previous responses that the oracle has given with respect to the j - 1 queries.

So that is why we can now safely conclude that the probability that hi = hj happens where hj is now defined like this by adversary making an oracle call for the inverse function is upper bounded by 1 over the size of x set - j + 1 right. So in both the cases that means irrespective of whether the jth queries for the Fk function or the jth queries for the Fk inverse function, the probability that the hash value which adversary learns due to this jth query matches the output of the ith query is upper bounded by 1 over the size of x-j+1, right. So that is the fact we have established now.

(Refer Slide Time: 28:58)



How many queries adversary is actually making here? We are making an assumption that adversary is making total q number of queries, total right and what we have analyzed is that out of those q queries, what is the probability that a pair of distinct queries gives him a collision for the Davies-Meyer construction in the ideal-cipher model, that probability we have calculated just now. So now what we have to do is we have to apply the union bound, namely we have to take this probability and sum it up over all pairs of distinct i, j where i and j ranges from 1 to q.

So let collision be the event that by making q number of queries to the family F and F inverse, adversary obtains at least one pair of collision that means adversary obtains at least one i, j pair where the hi value and hj values is same. So it turns out that by applying the union bound, the probability of collision can be easily upper bounded by this, namely summation of j = 1 to q and internally summation i = j - 1, the probability that the ith output and the jth output are same.

What I can do is that, I can substitute the inner summation because now I know that the value of each of the probabilities that is present in the internal summation is upper bounded by 1 over x – j + 1, and since i ranges from j-1 and there are j-1 such terms, so in the numerator we get j-1 and each of those probabilities is 1 over x- j + 1. So that is what we obtain and if we further simplify, then we can replace this inequality by this, namely what we have done here is I can always replace each of this - j +1 by - q because j is always upper bounded by q.

If I take out this summation finally since j ranges from 1 to q, I can replace inequality by the inequality by this inequality. So now let us analyze this final inequality for 2 cases. If the number of queries which the adversary has made in this ideal-cipher model is upper bounded by the size of x over 2, then we can do some simplification here and end up showing that the probability of collision is indeed upper bounded by q times q + 1 over the size of x set.

On the other hand if the number of queries what the adversary has made is greater than the size of x over 2, then trivially it so holds that probability of collision is upper bounded by this. That means it does not matter whether we are in case 1 or in case 2, in both the cases the probability of collision by making q queries is upper bounded by q over q + 1 over the size of x and that establishes the fact that we have stated in this theorem.

That means if q is say some polynomial function in the security parameter and if x is say some significantly large quantity, say some exponentially large quantity in the security parameter, then overall this probability turns out to be a negligible function in the security parameter and that is why we can now safely conclude that the Davies-Meyer construction is collision-resistant in the ideal-cipher model and adversary by evaluating the Davies-Meyer function for q number of messages in the ideal-cipher model could not come up with a collision except with a negligible success probability, right

So that brings me to the end of this lecture. Just to summarize in this lecture, we had discussed how to construct the fixed-length compression function and there are 2 approaches for that, one approach where we actually use number-theoretic hardness assumptions and we get provably secure guarantees in the standard model, but we do not opt those constructions in practice because the amount of computations which are involved in those constructions are of order of several magnitude.

Rather we use constructions based on dedicated block ciphers or the existing block ciphers and we had seen one of those construction namely the Davies-Meyer construction which is very simple, but it turns out that even though practically no attacks have been reported on that construction, we cannot prove the security of that construction just assuming that your block cipher is a secure site block cipher or a strong pseudorandom permutation.

We have to make very strong assumptions in our model, namely we have to assume that we are in the ideal-cipher model where block cipher is basically acting as a collection of or is a family of several truly random permutations where the access to the block cipher is through oracle calls even for the adversary. Only if we make this ideal-cipher assumption model, we can prove the security of the Davies-Meyer construction. Thank you.