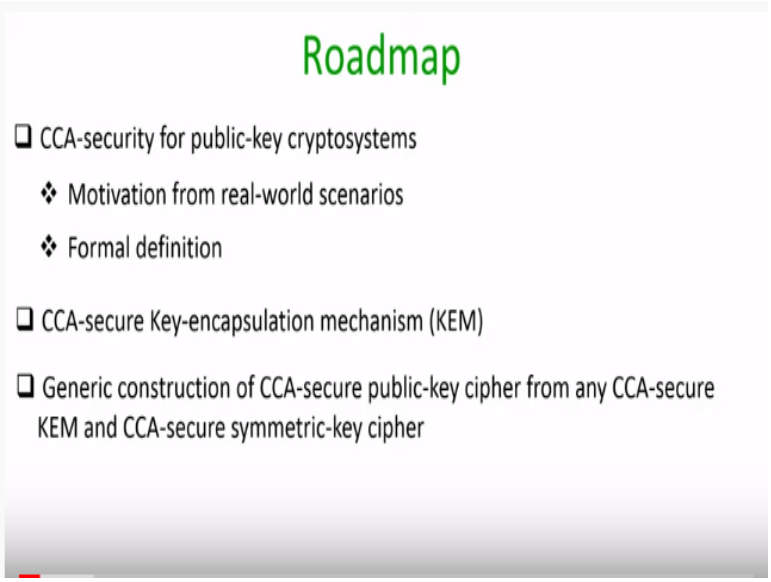
Foundations of Cryptography
Dr. Ashish Choudhury
Department of Computer Science
Indian Institute of Science – Bangalore

Lecture – 48
CCA Secure Public Key Ciphers

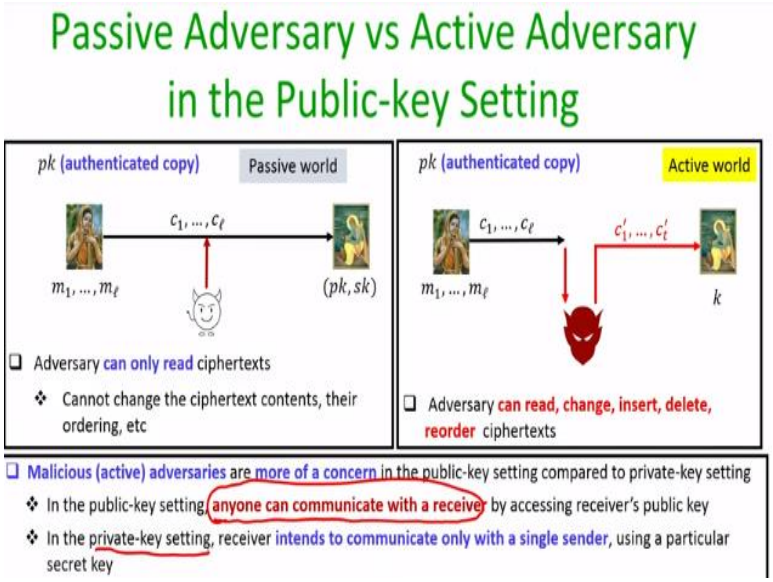
(Refer Slide Time: 00:34)



Hello everyone, welcome to this lecture just to recap in the last to the last lecture we have discussed rigorously the notion of CPA security in the context of public encryption schemes. So the plan for this lecture is as follows we will discuss the we will introduce the notion of CCA security for public key cryptosystems. We will discuss the motivation for studying CCA security considering real world scenarios.

We will see the formal definition of CCA security Then we will see the CCA security definitions for key encapsulation mechanism and then we will see a generic construction of CCA secure public key cipher from any CCA secure key encapsulation mechanism and any CCA secure symmetric-key cipher.

(Refer Slide Time: 01:13)



So let us begin our discussion with the difference between the passive adversary model and an active adversary model in the context of public key setting. So if we consider the passive adversarial model if we consider the passive adversary model then the scenario is the following So imagine we have a receiver here who has been done the key generation and he has established set up its public key made it available in the public domain and we assume that its public key named with authenticated copy of the receivers public keys available in the public domain.

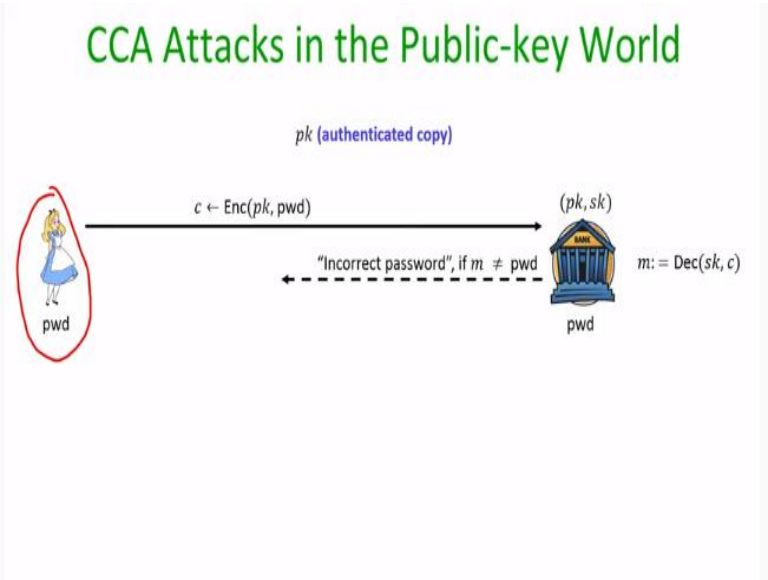
And using it say a sender encrypts a sequence of messages say m1 to ml and the resultant cipher text are communicated over the channel. Then in the passive model we assume that the adversary has the capability to only read the ciphertext it cannot change the ciphertext contents, it cannot change their ordering, it cannot introduce new ciphertext and so on. Whereas if we go in the active adversarial model then the adversary is more powerful in the sense it can not only read the ciphertext, but it can change the cipher text contents

It can insert new ciphertext on its behalf or on pretending that as if they are coming from the sender it can delete the ciphertext communicated by the sender. It can reorder the ciphertext and it can do any other any kind of attack which you can think of right? So it is a more powerful adversarial model compared to the passive adversarial model and it turns out so just to recall in the we have seen the difference between the passive adversarial model and the active adversary model in the context of symmetric key setting.

So right now we are now doing the same discussion in the context of public key setting and it turns out compared to the private key setting malicious or active adversaries are more of a concern in the public key setting and this is because of the fact that in the public key setting anyone can compute or communicate with a receiver just by accessing receivers public key because in the public key setting the encryption happens using the public key of the receiver and since it is going to be available in the public domain.

If I have an adversary and I want to compute ciphertext and send it to the receiver, I can do that this is in contrast to the private key setting where if I am an adversary and I do not have the symmetric key which is available between the sender and the receiver then its very unlikely for me to come up with a valid cipher test and send it to the receiver on the behalf of the sender if I am using an authenticated encryption scheme. So that means malicious adversaries are really a more of a concern in the public key setting compared to the private key setting.

(Refer Slide Time: 03:56)

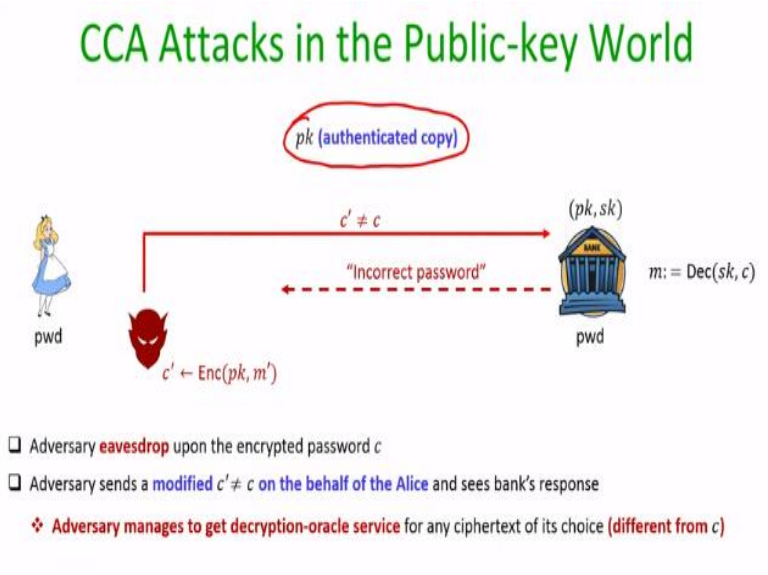


So that motivates us to study CCA attacks in the context of public key world. So what we are going to do next is we are going to see some real world scenarios where indeed CCA attacks can be launched. So consider this example where say a password is shared between a user and the bank and the banks public key the banks key public key is pk and its secret key is sk and its

public key pk is available in the public domain and the usual protocol between a legitimate user and a bank is as follows.

So if a user wants to initiate a session with the bank then the first thing the user does is it encrypts its password using the public key of the bank using some public key encryption process. And there is a certain cipher text is communicated or what the channel and receiving the encrypted password the bank decrypts a encrypted password and compares it with the password that it has stored with itself and it gives them an error message namely incorrect password. If it finds that on decryption that recovered password namely letter m is not the same as the password pwd which is stored at the banks site.

(Refer Slide Time: 05:15)



So this is a standard protocol so now let us see what happens if you take this simple protocol in the malicious adversarial model. So imagine we are given an adversary who is active who is an active adversary and who has used strop upon the encrypted password c it knows the public key of the bank because its available in the public domain but the adversary is not aware of the password which is encrypted in the cipher text c and its goal is to find out what exactly is the password.

Now what the adversary can do is since it knows the it can do is basically it can take the ciphertext c which has been communicated by encrypted password which has been

communicated by a legitimate user to the bank and it interrupts the communication and what it does it, it modifies certain bits of the cipher text c and come up with a new ciphertext c dash by encrypting some message m dash itself using the public key of the bank.

So what it is doing is it just stops the communication between the legitimate user and the bank. And instead it comes up with a new ciphertext to say c dash and c dash is an encryption of some known plain text m dash which is given to the which is known already to the adversary. And or maybe we can imagine that in this particular example the adversary can even do nasty things. In fact, it moves It can so happen that adversary not know m dash and it just modifies certain bits of the cipher text c and comes up with c dash and it might be the case that c dash is an encryption of him dash that is also a possibility.

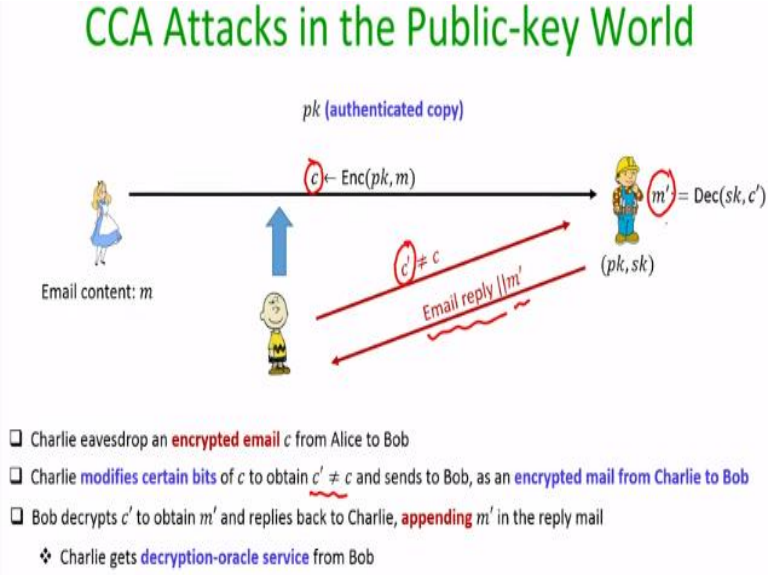
So whatever may be the case the adversary forwards the ciphertext to c dash to the bank and wait for the response of the bank. Now if it sees that in the response to c dash on decrypting c dash at visa the bank throws out the error message incorrect password. Then basically adversary here is actually coming to know that m dash is not the right password which was encrypted in c. Because if indeed m dash would have been the right password right which is encrypted in the ciphertext c then on decrypting ciphertext c dash bank will not have thrown the error message incorrect password.

But since the bank is throwing the message incorrect password somehow the adversary here get getting to learn that the password which is shared between the legitimate user and the bank is not m dash it is something different from m dash. And now the adversary can try to repeat the same attack again that means what it can do is it can just come up with another ciphertext say c double dash which could be an encryption of some plain text say m double dash and hope that indeed m double dash is the right password and forwards the cipher text c double dash to the bank and wait to see the banks respond.

And again if the error message come then the adversary learns here that the password is not m double dash and so on. So what is happening here basically in this example is that adversary is somehow getting a decryption oracle service from the bank without actually letting the bank

know that it is adversary who is actually persuading the bank to decrypt ciphertext of adversary's choice

(Refer Slide Time: 08:42)



Now let us consider another application here and here in this application say we have a you receive a Bob who has set up its public key and secret key and the public keys available in public domain and say Alice as a email say m which it wants to seek a communicate secretly to Bob. So what it does is it runs to public key encryption algorithm using the public key of Bob and the result tent encrypted mail c is communicated to both.

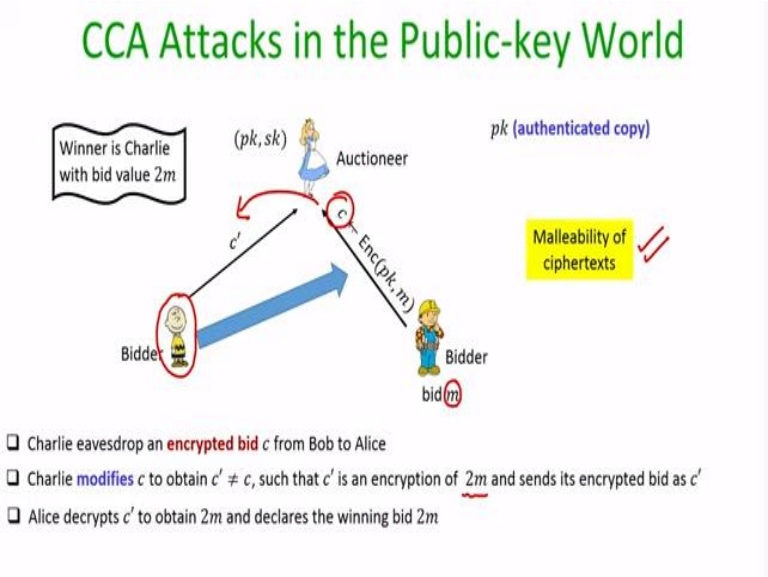
Now suppose Charlies interested to find out what is happening what exactly is the what exactly are the contents of the email? So what Charlie can do it is it can eavesdrop the encrypted email and what it can do it, it can modify certain bits of C to produce a new encrypted mail say c dash and send it to Bob and pretend as if that is an encrypted mail which Charlie would like to send to Bob right.

Now Bob what is going to do is when it receives the ciphertext c dash it will think as if Bob as if Charlie wants to send an encrypted email to Bob and on decryption what Bob can do is probably suppose on decrypting c dash it recovers the email content m dash and it might be the possible that Bob would like to reply back to Charlie and while replying back it might want to go to the message or the email which Bob has obtained after decrypting the encrypted email c dash.

That means the email reply which now Bob is sending might be concatenated with m dash depending upon the underlying application. Now when this response from Bob along with the decrypted email m dash comes back to Charlie what basically Charlies getting here is its getting a decryption oracle service namely it learns that the modified ciphertext c dash actually encrypts the email content m dash and in this case actually if indeed my encryption process would have been secure CCA secure then this this should not be possible right?

But since my encryption process is not CCA secure here Bob here on receiving a modified site protects to c dash its completely clueless that the modified email has been forwarded by an adversary here and its simply decrypting that modified ciphertext and responding back to the adversary thinking that the email originated from that person.

(Refer Slide Time: 11:15)



Now let us see the final example here and this is you can imagine a bidding protocol here and the scenario is the following. We have an auctioneer who has its public key set up available in the public domain and say we have 2 bidders who are bidding for a valuable object and say the bidder Bob goes first it has a private bid little m which it encrypts using the public key of the auctioneers with the auctioneer in this case is Alice here.

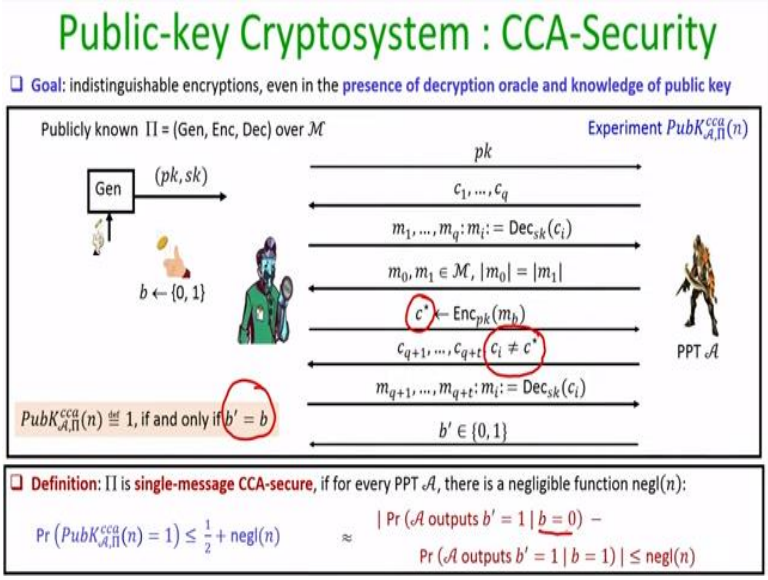
And now assume that Charlie is a malicious bidder who wants to win the bid, but it does not know the value little m because that is encrypted using the public key of the Alice. So what Charlie can do here is it can eavesdrop upon the bit encrypted bit of Bob and after doing that it can modify the interrupted bid c to another bit c dash and suppose my encryption process is such that that the modified encrypted bid c dash is an encryption of the bit 2 times the bid of bob and forwards the encrypted bid modified encrypted bit c dash to Alice pretend and Charlie pretend as if c dash is the bid which Charlie would like to make here.

So this property here where its possible for an adversary namely malicious Charlie to eavesdrop a ciphertext of an unknown message and from that ciphertext produce another related ciphertext c dash which is an encryption of some related message namely 2 times to message m which was encrypted in the ciphertext c is called as the malleability property of ciphertext. So recall we had in the when we were discussing symmetric encryption in process.

There also, we discussed the notion of malleability and malleability could be possible even in the context of public encryption process. So now in this example if indeed its possible for Charlie to convert c to c dash such that c dash is a good is a public incorruption of the message 2 times m then what Alice might do is when she decrypts c and c dash she will find that c dash is the winning bid.

Because it corresponds to the value 2 times m and she cannot publicly announce that Charlie has won the bid by bidding for the bid value 2 times m and in this case after learning the result Charlie ends up getting a decryption oracle service from Alice and ends up winning the auction which should have been avoided if indeed my encryption process would have been CCA secure right.

(Refer Slide Time: 13:55)



So now we have seen several real world scenarios where CCA attacks can be launched where the adversary can get the decryption oracle service. So hence now we have to study formerly the notion of CCA security in the context of public key crypto system So let us find one define that So on a very high level the goal of CCA security is to achieve indistinguishable encryptions even in the presence of decryption oracle and the knowledge of public key being available with the adversary and this is modeled by a challenge response KEM.

The rules of the KEM are as follows the challenger advances the key generation algorithm gives the public key to that adversary who is computationally bounded. So since the public is given explicitly to the adversary it can get encryption oracle service on its own by encrypted any plain topic of his choice. Now what it can do in this experiment here is it can ask for the decryption oracle service by submitting several cipher text from the cipher text base and in response the challenger has to decrypt back all those ciphertext using the secret key escape which is not known to the adversary.

Now the challenge phase starts where the adversary submits a pair of plain text on the only restriction being that their launch should be same and to prepare the challenge ciphertext our challenger randomly picks one of those messages and encrypts it using the public key pk and now we give that adversary access to the post challenge decryption oracle service where again it can ask for decryption oracle service or decryption for every for a mini ciphertext of its choice

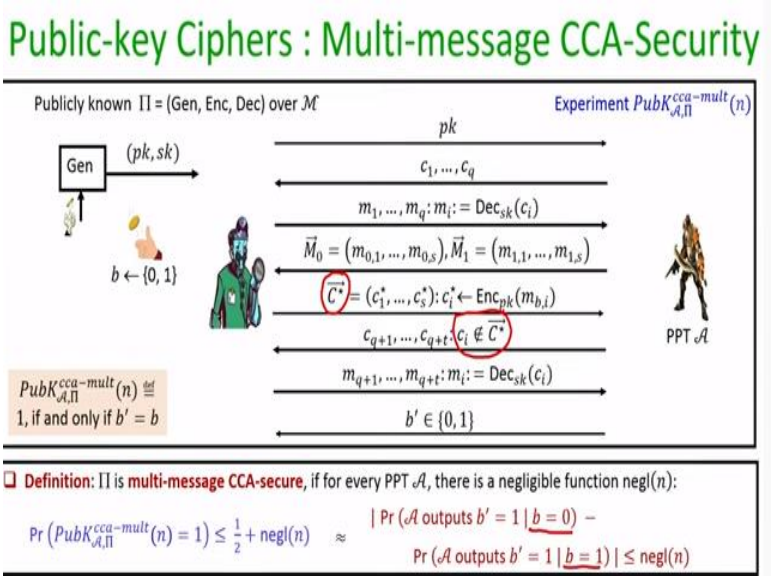
with the only restriction being that, that is post challenge decryption oracle service is restricted from being its adversary is restricted to ask for the decryption of the ciphertext c star.

Because if we do not put this restriction then we cannot achieve any meaningful notion of secrecy and if you look the 3 motivating examples that I given earlier in all those 3 examples the goal of the Charlie or the bad guy was to get a decryption oracle service of a modified ciphertext but not for the ciphertext which it is interested to crack. Now once the adversary gets a decryption oracle service for the post challenge decryption oracle queries the adversary goal is to identify where the c star is an interruption of m0 or m1.

So its submits, it is a response or output and the rule of the experiment is we say that the adversary has won the experiment which equivalent to saying that the output of the experiment is 1 if and only if b dash = b. That means adversary has correctly identified what is encrypted in challenge ciphertext and our security definition is we say that our encryption processes single message CCA secure if for every poly time adversary the success probability of adversary winning the KEM is upper bounded by some half plus negligible function in the security parameter or equivalently the distinguishing advantage of the adversary is upper bounded by some negligible function namely it does not matter whether c star is an encryption of m0 or m1 with almost same probability the response of the adversary will be same.

The reason we are calling this definition a single message CCA secure because adversary has just submitted a pair of challenge plain text and is seeing an encryption of one of them.

(Refer Slide Time: 17:13)



We can extend this definition in a straightforward version or the natural way to incorporate multi message CCA security. The rules of the KEM will be almost same as for the single message CCA security where challenger throws the public key to the adversary. adversary gets decryption oracle service and now in the challenge phase its allowed to submit a pair of vector of messages. But the only restriction being that component wise the message in the 0th factor in the message in the first vector should be off same length.

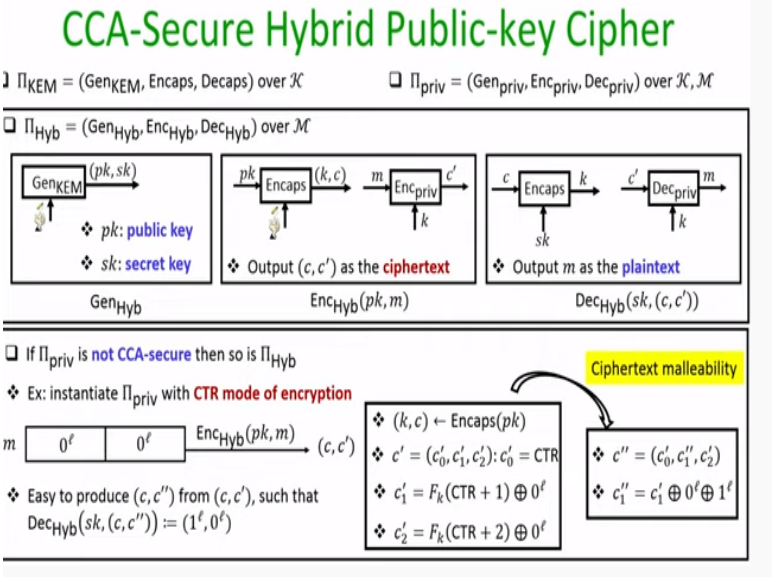
To prepare the challenge cipher text adversary the challenger picks one of these 2 vectors with equal probability for encryption and then encrypts all the plain text on those in the selected vector and the challenge ciphertext vector is given to the adversary, the adversary is again is now allowed to have access to the post challenge decryption oracle service with the only restriction that it cannot ask for the decryption of any ciphertext which is present in the challenge ciphertext vector.

Now once our adversary is sufficiently trained it has to identify whether the challenge ciphertext vector it has seen corresponds to an encryption of the 0th vector or the first vector. And we say that that was already has won the experiment or the output of the experiment is 1 if and only if it has correctly identified whether it is m0 or whether it is m1 which is encrypted in the challenge ciphertext vector c star.

And our security definition is we say that our encryption processes multi message CCA secure if for any poly time adversary participating in this experiment the probability that it can win the experiment is upper bounded by half plus some negligible function in the security parameter or equivalently the distinguishing advantage of the adversary is upper bounded by some negligible function in the security parameter.

And as expected we can prove that single message CCA security and multimedia CCA security are equivalent even in the context of public key encryption schemes. So I am not giving the full proof here you can refer to the book by (()) (19:23) for the full proof.

(Refer Slide Time: 19:25)



So now let us define the notion of CCA security in the context of hybrid public key ciphers. So remember in the last lecture we have discussed that how using a key encapsulation mechanism and a symmetric encryption scheme we can come up with a combination of both of them to come up with a more efficient hybrid encryption process right? Where the key generation algorithm of the hybrid encryption process will run the key generation algorithm of the KEM and output.

The public key and secret key where secret key will be available with the receiver and public key will be available in the public domain. The encryption process of the hybrid scheme will be as follows it runs first a key encapsulation algorithm and obtains a symmetric key little k and an

encapsulation of the key will de denoted by little c and then the key little k is used to encrypt the plain text according to the symmetric key encryption algorithm to produce the cipher text c dash.

And the overall ciphertext is the encapsulation of the symmetric key and the encapsulation of the plain text and analog of c the decryption happens at the receiving end the receiver first de capsulate the encapsulation c and obtains the symmetric key little k and once it obtains the symmetric key little k it decapsulates the ciphertext component c dash to recover back the plain text little m using the decryption algorithm of the symmetric encryption process.

And also to recall in the last lecture we have proved that if my KEM is CPA secured and my symmetric encryption process is COA secure then the overall scheme is CPA secured. But since now we are considering CCA security we have to identify what should be the security properties on my underlying building blocks. It turns out that if I want to achieve CCA security then definitively my underlying symmetric encryption process in the hybrid encryption scheme should be CCA secure.

It is not suffice to just have a COA security or CPA security for the underlying symmetric encryption process to demonstrate my point let us instantiate the underlying symmetrical block here in this hybrid encryption process by a counter mode of operation which we know is CPA secure but not CCA secure. So imagine that a sender has encrypted a message consisting of 2 blocks off all 0s using the hybrid encryption scheme as per the above hybrid encryption process and the result in ciphertext to c, c dash and as per the details of the encryption process of this hybrid encryption scheme the way c, c would have been produced is as follows.

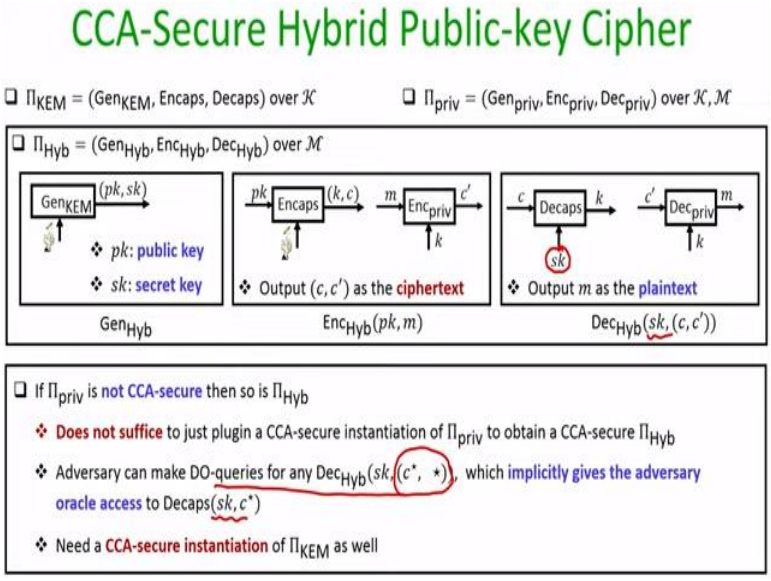
First an encapsulation algorithm would have been executed to obtain a symmetric key and the encapsulation of that key and then using the key k by invoking the counter mode of operation the message block all 0s followed by all 0s would have been executed. So the encryption of the message the little m using the key as per the counter mode of operation will be as follows. The random counter will be selected which will be available as part of the cipher text component c dash and actual encryption of the blocks of the message will be c1 dash and c2 dash as per this counter mode of operation.

Now what we are going to see here in the example is that imagine there is an adversary who has eavesdropped the ciphertext c, c dash and imagine that the adversary as an active adversary then by observing c, c dash it is very easy to follow that adversity to produce a modified cipher text c, c double dash such that when this modified cipher text is forwarded to the receiver and decrypted as per this hybrid encryption process it leads to the plain text all 1s followed by all 0s and the way adversary can do that is by exploiting the malleability of the counter mode of operation.

Basically he has to produce c double dash which we are the counter value namely c, 0 dash is retained as it is and the c2 dash component of the ciphertext c dash is also retained as it is. The modification is only in the ciphertext component c1 dash, c1 dash is now changed to c1 double dash as follows and if c1 dash has changed to c1 double dash has follows then the effect of all 0s and all 0s cancels out.

And basically c1 double dash now corresponds to a counter mode of operation for the message block all once and now this since this overall process is malleable we can easily show that this is not going to be CCA secure. So that means if at all we want the overall hybrid encryption process to be CCA secure definitively my underlying symmetric encryption scheme which I am using should be CCA secure.

(Refer Slide Time: 24:20)



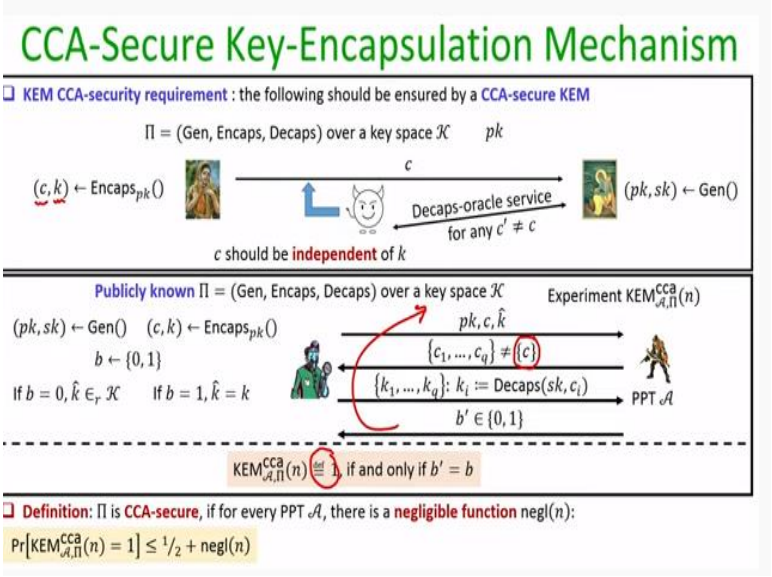
But it turns out that just instantiating the underlying symmetric encryption by a CCA secure symmetric encryption does not suffice to give us an overall CCA secure hybrid encryption process and the reason for this is if you play the CCA KEM against this hybrid encryption process then remember that in the CCA KEM adversary is given access to the decryption oracles service.

Namely the adversary can now make decryption oracle service for any kind of modified cipher text where the first part of the ciphertext can be any c star namely any encapsulation followed by anything and to respond to those modified queries modified decryption oracle queries the challenger has to basically decrypt such modified cipher text using the secret key escape.

Now if you see the decryption algorithm of this hybrid encryption scheme any decryption oracles query of the form c star followed by anything when it gets decrypted by the challenger in the CCA KEM basically it provides implicitly that adversary oracle access to the decapsulation oracle service under the unknown secret key escape. Because on decrypting the modified ciphertext adversary will come to learn what exactly the decapsulation of start under the unknown secret key sk corresponds to.

That means we now need a CCA secure instantiation of the key encapsulation mechanism as well to hope that the overall hybrid encryption process results in a CCA secure public encryption process.

(Refer Slide Time: 26:01)



So let us first define the notion of CCA security for key encapsulation mechanism and on a very high level the goal of a CCA secure key encapsulation mechanism should be is to ensure the following. So imagine we have a receiver who runs the key generation algorithm of a CCA secure encapsulation mechanism and set up the public key and say our sender is there which runs the key encapsulation algorithm of that scheme obtains a secret key sk and the encapsulation of the key little c and encapsulation is sent to the receiver and say there is a malicious adversity who has eavesdropped the encapsulation c.

And now imagine that my adversary gets the decapsulation oracle service for any encapsulation c dash different from c. Now by getting polynomial number of decapsulation oracle service we required that my adversary from the viewpoint my adversary the encapsulation c which it has seen earlier should we still independent of the key k which is encapsulated in the c right. So the advantage here that my adversary is now getting is that its now getting a implicit explicit decapsulation oracle service which we have now to model in our experiment.

So to model above requirement of experiment is as follows double challenger it runs the key generation algorithm and using the public key it runs the encapsulation algorithm to obtain a pair c, k and now it prepares the challenge for the adversary as follows it tosses a fair coin if the coin tosses 0 then it picks a random element k hat from the key space. Whereas if the coin tosses 1

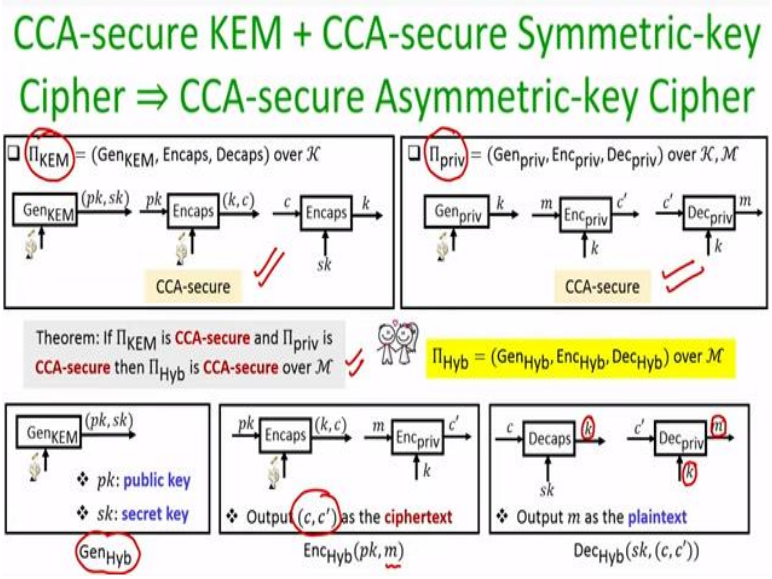
then the element k hat is the key k which is actually encapsulated in the encapsulation c and the challenge for their adversary is as follows.

The public is given the encapsulation c is given and k hat is given, and the goal of the adversary is to identify whether k dash is a random element from the key space namely whether b = 0 or whether k dash is the same key which is encapsulated in c namely b = 1. But now we model that the capsulation oracle service by allowing that verse to ask for decapsulation of any encapsulation of its choice with the only restriction being that this decapsulation oracle service should to not be for the encapsulation c.

They should be different from c and our adversary is allowed to adaptively submit its query and in response to the decapsulation oracle queries the challenge or response by decapsulating all those queries under the unknown secret key sk which is not known to the adversary and after making polynomial number of queries the adversary has now has to identify and solve its challenge.

Namely it has to identify whether he has seen a challenge as per method it will be equal to 0 or as per the method it will be equal to 1 and the definition of the experiment is we say that adversary has won the experiment which we denote by saying that output of the experiment is 1 if and only if adversary has ensured b dash = b and we say that our key encapsulation mechanism is CCA secure if for every poly time adversary there exists some negligible function such that the probability of that adversary winning the experiment is upper bounded by half plus negligible function or equivalently the distinguishing advantage of that adversary is upper bounded by some negligible function in the security parameter.

(Refer Slide Time: 29:42)



So now we are going to see that if we are given a CCA secure KEM and a CCA secure symmetric key cipher then if we combine it, we get us CCA secure asymmetric key cipher. So imagine we are given a CCA secure KEM and the CCA secure symmetric key encryption process then we can combine it in the same way as we have done to obtain a CPA secure asymmetric decipher in the last lecture.

So my key generation algorithm of the hybrid encryption process will be simply the key generation algorithm of the key encapsulation mechanism to encrypt a plain text using the public key pk what sender is going to do is it will run the encapsulation algorithm and will obtain a key k and its encapsulation c and using the key k it will invoke the encryption algorithm of the underlying symmetric encryption process to encrypt the plain text little m and obtain its encapsulations c dash and the overall cipher text will be see c, c dash.

On the other hand, the receiver who possessed the secret key sk on the receiving the ciphertext to c, c dash will first decapsulate the c part of the ciphertext to retrieve the encapsulated key little k and then that little k then the key little case used to decapsulate the c dash component of the cipher text as per the decryption algorithm of the symmetric encryption process to get back the actual plain text little m.

And we can prove that if my key encapsulation mechanism is CCA secure as per the definition that we have just given and if my underlying symmetric encryption processes is CCA secure then this generic way of combining these 2 primitives is going to give us a public key encryption process which is CCA secure. And the proof again will be something similar to the hybrid argument style proof which we had given in the last lecture to prove the CPA security of the generic construction of the hybrid scheme that we had discussed in that lecture

So I am leaving the full formal details of the proof to you as an exercise. So that brings me to the end of this lecture just to summarize. In this lecture we introduced the notion of CCA security in the context of public encryption scheme. We have seen the malleability what exactly malleability of public encryptions schemes means and we have seen the definition of CCA security for key encapsulation mechanism and discuss that if we are given a CCA secure key encapsulation mechanism and a CCA secure symmetric decipher then we can combine them generically to obtain a hybrid encryption process which is public key hybrid encryption process and CCA secure. Thank you.