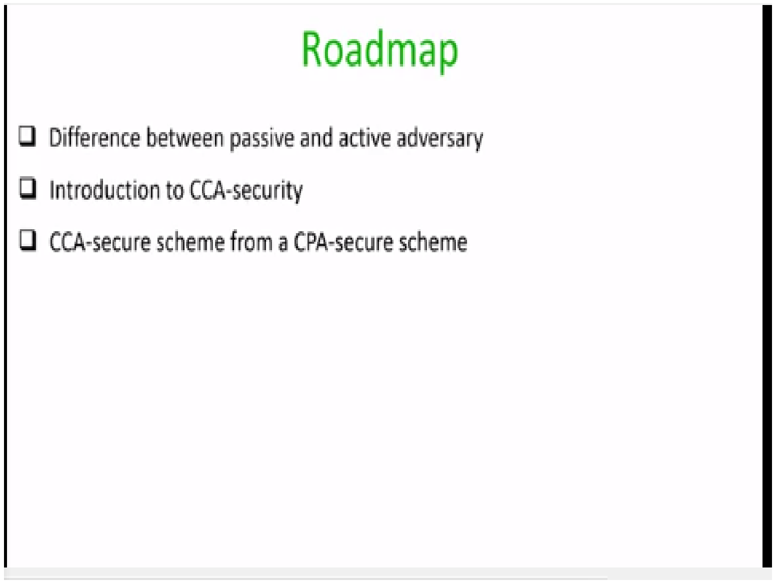
Foundations of Cryptography

Prof. Dr. Ashish Choudhury
(Former) Infosys Foundation Career Development Chair Professor
International Institute of Information Technology-Bangalore

Lecture-21
From Passive to Active Adversary

Hello everyone, welcome to lecture 20, so just a brief recap.

(Refer Slide Time: 00:34)



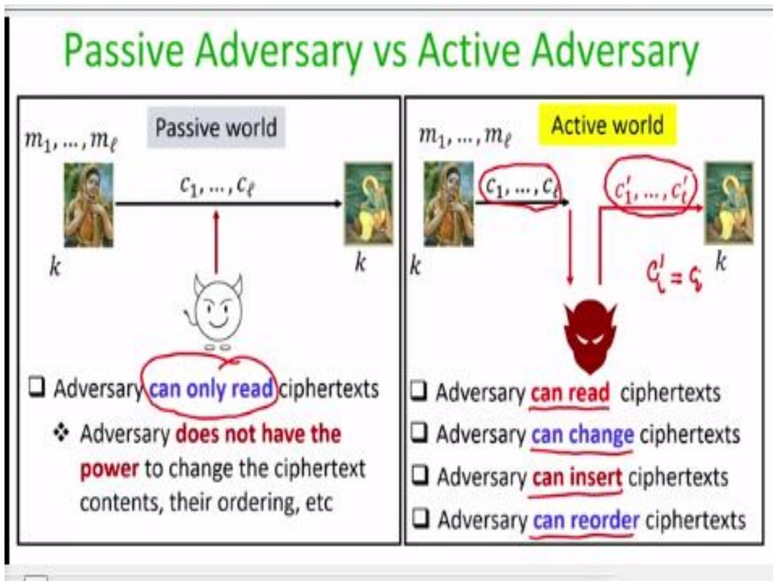
Roadmap

- ❑ Difference between passive and active adversary
- ❑ Introduction to CCA-security
- ❑ CCA-secure scheme from a CPA-secure scheme

Till now we have discussed about symmetric key cryptography in the presence of a passive adversary or an eavesdropper who can just eavesdrop the communication happening between the sender and a receiver. But from now onwards we will consider a more powerful adversary, namely an active adversary. And we will see what exactly are the harmful effects of the presence of an active adversary.

More specifically, the roadmap for this lecture is as follows, we will introduce active adversary and we will discuss the differences between a passive adversary model and an active adversary model. And we will also introduce the notion of CCA security. And we will discuss how exactly to approach designing CCA secure schemes from a CPA secure scheme.

(Refer Slide Time: 01:20)


Passive Adversary vs Active Adversary

- ❑ Adversary can only read ciphertexts
  - ❖ Adversary does not have the power to change the ciphertext contents, their ordering, etc
- ❑ Adversary can read ciphertexts
- ❑ Adversary can change ciphertexts
- ❑ Adversary can insert ciphertexts
- ❑ Adversary can reorder ciphertexts

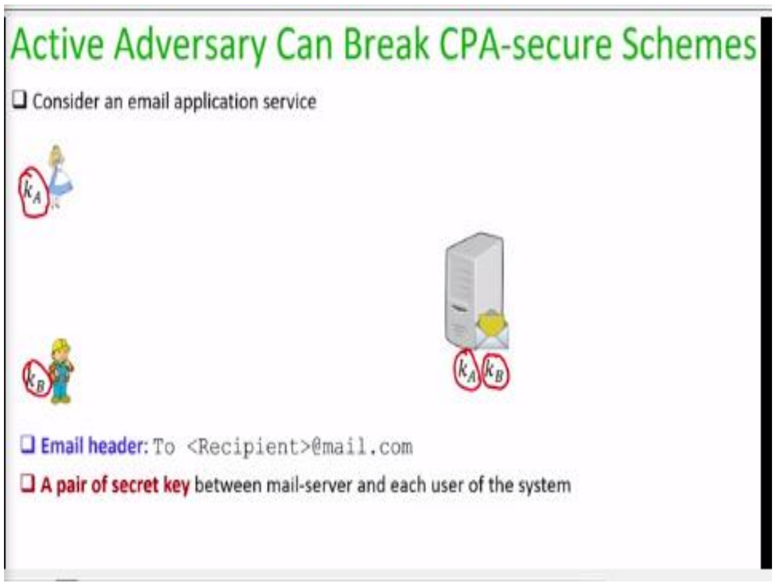
So let us start our discussion by discussing the differences between the passive adversarial model and the active adversarial model. So in the passive adversarial model, the scenario is the following we have a sender and a receiver with a shared key agreed upon by some magical mechanism. And say sender has a sequence of messages which it has encrypted by using some encryption algorithm which is publicly known and a ciphertext are communicated over a publicly known channel.

Where an adversary can eavesdrop and read the ciphertext right. So in the passive adversary model, the adversary can only read the contents of the ciphertext. And adversary does not have any power to change the ciphertext contents, to reorder them, to insert new ciphertext of its own, to delete some ciphertext etc. That means the only capability of the adversary in the passive adversarial model is the reading capability.

Whereas if we go to the active adversarial model, and the scenario is the same with respect to the sender and the receiver. Namely, pre-shared key is agreed upon between the sender and the receiver. Sender has a sequence of messages encrypted by some encryption process. And now we assume that we have a powerful adversary who is active. And what exactly I mean by active adversary is that it can not only read the ciphertext communicated by the sender.

But the adversary is allowed to change the contents of the ciphertext. It can insert new ciphertext of its own, or it can reorder the ciphertext sequence as the sent by the sender. So that is why the number of ciphertext is here and the ciphertext which are actually communicated by the sender, they are denoted by c_1 to c_l. Whereas the ciphertext or the bit strings which are actually received by the receiver they are denoted by c'_1, c'_2, like that, c'_t, where t could be different from *l* and each c_i' may not be equal to the corresponding c_i and so on right. So, that means in the active adversarial model, we have a more powerful adversary who is more powerful compared to the adversary in the passive adversarial model.

(Refer Slide Time: 03:43)



Active Adversary Can Break CPA-secure Schemes

- Consider an email application service

- Email header: To <Recipient>@mail.com
- A pair of secret key between mail-server and each user of the system

So, now what we are going to illustrate is that as soon as we consider a more powerful adversarial model namely an active adversary, whatever encryption schemes that we had seen till now namely whatever candidates CPA secure schemes we have discussed till now, all of them can be broken in the presence of an active adversary. That does not mean that the encryption schemes that we have discussed are insecure.

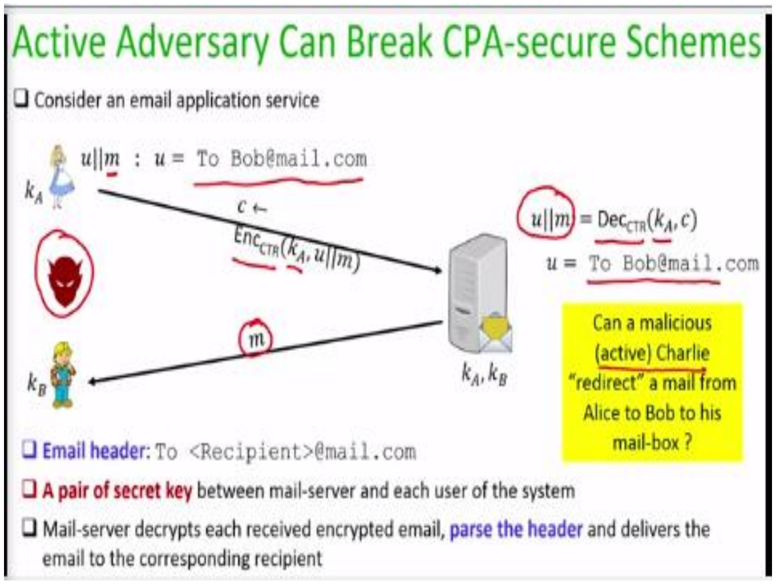
They are secure in a weaker adversarial model, namely a passive adversarial model where the adversary is restricted only to eavesdrop. But as soon as we go to a more powerful adversarial model namely active adversarial model, our goal of this illustration is to show how an active adversary can break the CPA secure scheme.

So we consider an email application service and we assume that we have a mail server whose domain name is mail.com. And we assume that each email header the first part of the email header is the identity of the receiver. That means the email header will have the format “To” followed by the recipient name at mail.com. So that will be the syntax of the email header in this application.

And we assume that in this application, all the users who are using the service of mail.com, it has a pair of secret key pre shared and secretly available to the user under corresponding mail server. So for example Alice will have pre set k_A, which is available to Alice and to the main server and not known to anyone else. In the same way we assume another user of the system, say Bob has a secret key k_B which is shared between the mail server and Bob.

And like that, if we have n number of users, we assume that each user has a secret key, which is shared between that specific user and a mail server.

(Refer Slide Time: 05:50)



Active Adversary Can Break CPA-secure Schemes

- Consider an email application service

$u\|m$: $u =$ To Bob@mail.com
k_A
$c \leftarrow Enc_{CTR}(k_A, u\|m)$
$u\|m = Dec_{CTR}(k_A, c)$
$u =$ To Bob@mail.com
m
k_B
k_A, k_B
Can a malicious (active) Charlie “redirect” a mail from Alice to Bob to his mail-box ?

- Email header: To <Recipient>@mail.com
- A pair of secret key between mail-server and each user of the system
- Mail-server decrypts each received encrypted email, parse the header and delivers the email to the corresponding recipient

So that is the setting we are assuming here and the way this application works is as follows. You can imagine that any encrypted mail which comes to the email server, the email server decrypts that email using the key of the corresponding user from which or from the corresponding sender from which the email is coming. And then once the email is decrypted, then the email server passes the header from which it learns the recipient of that email.

And hence, the corresponding email is forwarded to that particular recipient. So for instance, if Alice is interested to send a mail, say message m or a mail m to Bob. Then Alice will prepare a plain text where the first part of the plain text will be the identity of the recipient. In this case, the content of u will be the binary representation of the string “To Bob@mail.com”, followed by whatever emails she wants to communicate to Bob.

So that will be the plain text of the Alice if she wants to send an email to Bob. And what Alice is going to do is it is going to encrypt that email using some CPA secure encryption process. So we assume that we are using the counter mode of operation of block cipher or AES for instance counter mode of operation of AES to encrypt the email content where the secret key which is used to encrypt the message is the key k_A which is shared between Alice and the mail server.

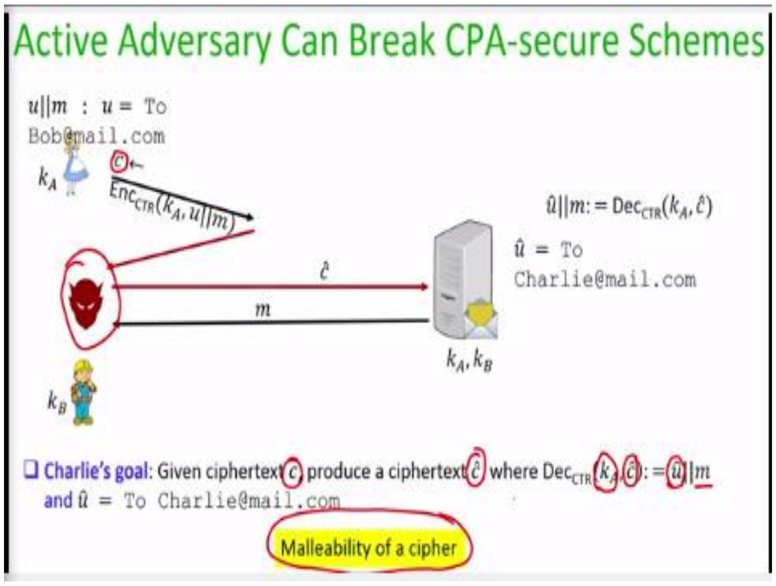
So that will be the encrypted mail which Alice will communicate to the mail server. Now once the encrypted email comes to the mail server, the mail server knows that the email is coming from Alice and it decrypts as per the counter mode of decryption. And the key which is used for decryption is the key which is shared between the mail server and Alice, namely k_A. And after decryption, the mail server recovers the corresponding plain text which it parses at the identity of the recipient followed by the mail content.

So in this case, the mail server finds that the recipient of the email is Bob, and hence it forwards the email to Bob. So I am assuming that the mail is forwarded as it is in clear to Bob but you can imagine to provide more privacy, Bob can further encrypt the email m using the key k_B using any CPA secure encryption process, so that is the way the system works. So now what we are going to show in this illustration is that, if we assume that our adversary is an active adversary.

That means we assume that we have a malicious user in the system who is active, and its name is say Charlie. So what we are going to see is that, is it possible for a malicious Charlie to redirect the mail from Alice to Bob, to its mailbox without actually Alice or the mail server, knowing about this fact, right. And we are assuming here that Charlie is an active adversary that means it is not only allowed eavesdrop the communication between Alice and the mail server.

But it can insert it is own ciphertext, it can reorder a ciphertext and so on. So we are no longer in the passive adversarial model. And our goal is to show that indeed it is possible for a malicious Charlie to redirect a mail from Alice intended for Bob to actually go to Charlie's email box, right.

(Refer Slide Time: 09:16)


$u||m$: u = To Bob@mail.com
k_A
c
$\text{Enc}_{\text{CTR}}(k_A, u||m)$
$\hat{u}||m := \text{Dec}_{\text{CTR}}(k_A, \hat{c})$
$\hat{u}$ = To Charlie@mail.com
$\hat{c}$
m
k_A, k_B
k_B

Active Adversary Can Break CPA-secure Schemes

- Charlie's goal: Given ciphertext c produce a ciphertext $\hat{c}$ where $\text{Dec}_{\text{CTR}}(k_A, \hat{c}) := \hat{u}||m$ and $\hat{u}$ = To Charlie@mail.com

Malleability of a cipher

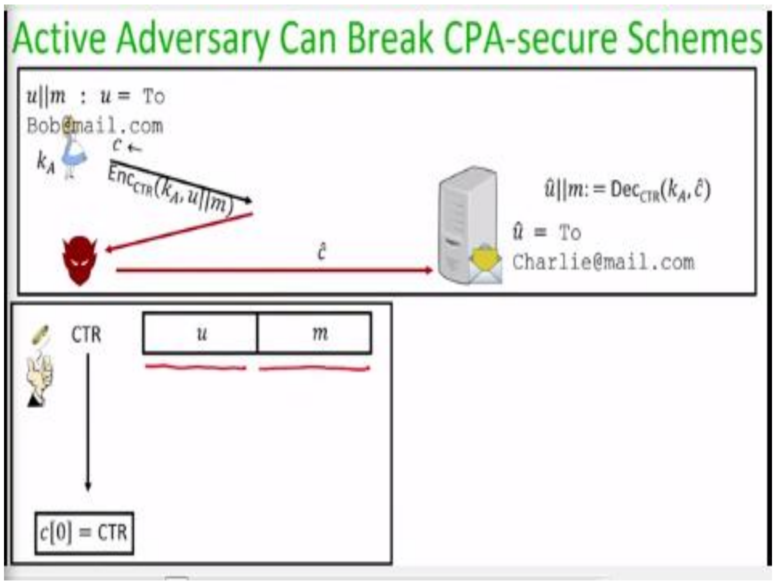
So, basically, the way system is operated is as follows just recall that Alice has encrypted an email, and the ciphertext was c which was communicated from Alice to Bob. And now what the malicious Charlie does is, it does the following. It intercepts the encrypted email which is going from Alice to the email server and the goal of the Charlie now is as follows. Once it intercepts the encrypted email, namely the ciphertext c.

The goal of the Charlie is to produce a new ciphertext which I denote by $\hat{c}$ and forward it to the mail server. So, that the modified ciphertext or the for ciphertext $\hat{c}$ which is now forwarded to the email server. When decrypted using the counter mode of operation using the key k_A produces an email produces a message content where the email content remains the same namely m as sent by the Alice.

But the recipient address namely $\hat{u}$ instead of Bob@mail.com, $\hat{u}$ corresponds to Charlie@mail.com. If our malicious attacker Charlie is able to produce this ciphertext $\hat{c}$ from the ciphertext c. Then when $\hat{c}$ is decrypted by the mail server, the corresponding email m will be forwarded to Charlie's email box instead of Bob's email box.

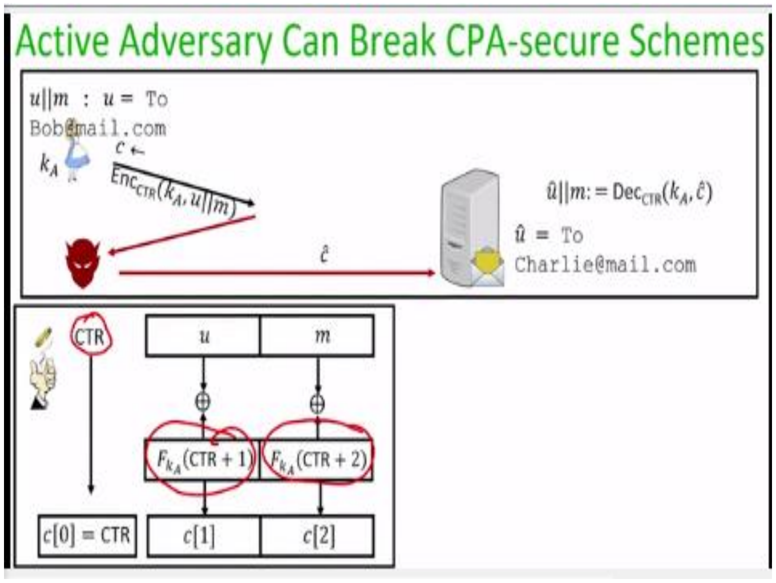
So that is what exactly is the goal of Charlie and if Charlie is able to do that, the goal of the Charlie is achieved, right. So, this feature of producing a related ciphertext $\hat{c}$ from an existing ciphertext. So that the underlying plain text underneath c and $\hat{c}$ are related is known as the malleability of a cipher. And if the ciphertext, if the encryption process that we are using has this malleability property, then indeed Bob will be able to successfully achieve its goal.

(Refer Slide Time: 11:30)


Active Adversary Can Break CPA-secure Schemes
$u||m$: u = To Bob@mail.com
k_A
$c \leftarrow \mathsf{Enc}_{CTR}(k_A, u||m)$
$\hat{c}$
$\hat{u}||m := \mathsf{Dec}_{CTR}(k_A, \hat{c})$
$\hat{u}$ = To Charlie@mail.com
CTR
u
m
$c[0]$ = CTR

So, let us see whether indeed it is possible for Bob to achieve its goal if we are using a counter mode of operation. So imagine that the email or the message which Alice has encrypted, it consists of the message block u and the actual email block say m. And for simplicity I am assuming that both u, namely the identity of the receiver which in this case is Bob@ mail.com. And actually mail block all of them fits in 2 consecutive blocks of the underlying pseudo random function which we are using in the counter mode of operation. This is just for simplicity but even if that is not the case, we can assume that the plain text u concatenated with m is encrypted as per the counter mode of operation.

(Refer Slide Time: 12:22)

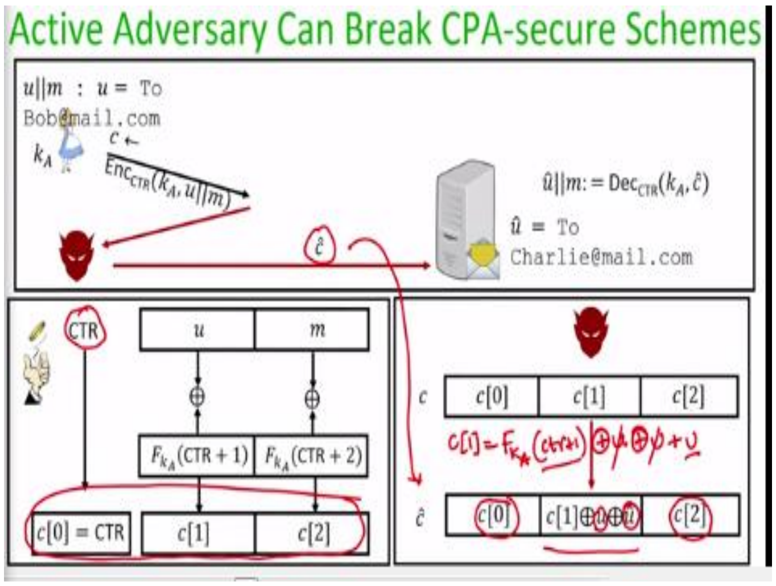

Active Adversary Can Break CPA-secure Schemes
$u||m$: u = To Bob@mail.com
k_A
$c \leftarrow Enc_{CTR}(k_A, u||m)$
$\hat{c}$
$\hat{u}||m := Dec_{CTR}(k_A, \hat{c})$
$\hat{u}$ = To Charlie@mail.com
CTR
u
m
$F_{k_A}(CTR+1)$
$F_{k_A}(CTR+2)$
$c[0]$ = CTR
$c[1]$
$c[2]$

So, the way counter mode of operation would have operated on this plain text will be as follows. Alice would have picked around random value of the counter and that will be the 0th ciphertext block. And since in this particular example, we are assuming that a plain text consist of 2 blocks, the ciphertext will consist of 2 blocks, namely c_1 and c_2, where c_1 will be obtained by first evaluating the underlying keyed PRF with the key k_A, at the block input CTR + 1.

And by evaluating the same keyed PRF with the key k_A, with the block input CTR + 2. And the resultant outputs are used as the pads for masking the message blocks u and m and that gives you the corresponding ciphertext block c_1 and c_2. So, basically what it means is that c_1 is the XOR of the message block u with the value of the keyed PRF at this counter value. And a ciphertext block is basically the XOR of the message plain text content m with the PRF output and at the input CTR + 2.

And our malicious Charlie is aware of this fact because it knows the encryption process which is used by Alice to produce the ciphertext c.

(Refer Slide Time: 13:45)


Active Adversary Can Break CPA-secure Schemes
$u||m$: u = To Bob@mail.com
k_A
$c \leftarrow \mathrm{Enc}_{CTR}(k_A, u||m)$
$\hat{c}$
$\hat{u}||m := \mathrm{Dec}_{CTR}(k_A, \hat{c})$
$\hat{u}$ = To Charlie@mail.com
CTR
u
m
$F_{k_A}(CTR+1)$
$F_{k_A}(CTR+2)$
$c[0] = CTR$
$c[1]$
$c[2]$
c
$c[0]$
$c[1]$
$c[2]$
$c[1] = F_{k_A}(ctr+1) \oplus u \oplus u \oplus \hat{u}$
$\hat{c}$
$c[0]$
$c[1] \oplus u \oplus \hat{u}$
$c[2]$

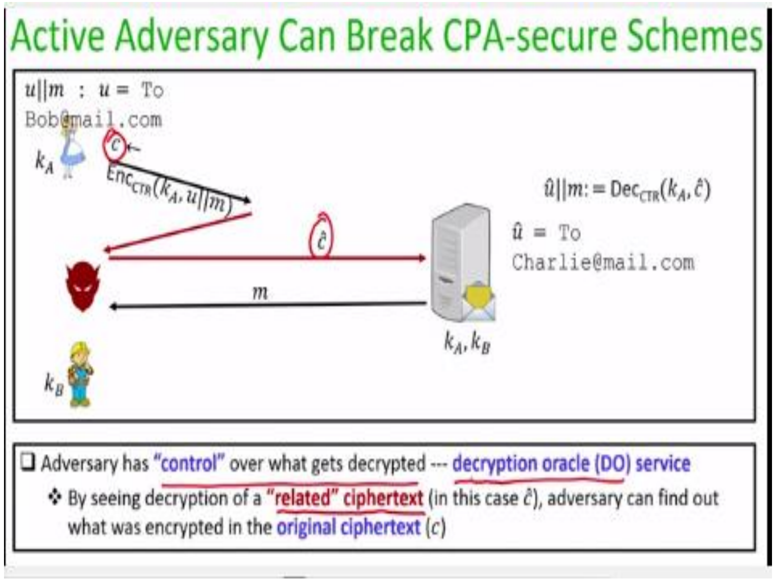
Now what is the goal of malicious Charlie, its goal is to basically take this ciphertext c consisting of 3 blocks c_0, c_1 and c_2 and produce a related ciphertext $\hat{c}$, satisfying the property that we had discussed earlier. So here is how the malicious Charlie can prepare the modified ciphertext $\hat{c}$, $\hat{c}$ basically consist of 3 ciphertext block where the 0th ciphertext block is the same as the 0 th ciphertext block of the original ciphertext.

Namely, the value of the counter, which was used by Alice is retained as it is, and a second ciphertext block, namely c_2 of the modified ciphertext and original ciphertext remains the same. Because the ciphertext block c_2 actually is the encryption of the plain text, which the malicious Charlie wants to be forwarded to Charlie's mailbox. So it does not want to mess up with the second ciphertext block in the modified ciphertext.

But if you see the first ciphertext block in the modified ciphertext is set to the XOR of the first ciphertext block of the original ciphertext and the value of u XORed with $\hat{u}$. And as you can see that the value u as well as $\hat{u}$ are known to the malicious Charlie, u in this case is the string binary string corresponding to Bob@mail.com.

And $\hat{u}$ is the binary string corresponding to the string Charlie@mail.com. And if you see, if you do XOR of c_1 with u and $\hat{u}$ then since c_1 is actually an encryption of u, namely, it is a value of your keyed PRF at the counter value CTR + 1 XORed with u. And if you further XOR with u and $\hat{u}$, then the effect of u and u cancels out, so in a sense, the first block of the modified ciphertext now actually corresponds to the encryption of the string $\hat{u}$ under the counter value CTR + 1 as per the counter mode of operation. And if our malicious Charlie prepares c like this, and forward it to the mail server when the mail server decrypts this modified ciphertext thinking that it is actually coming from Alice. The mail server will decrypt it as per the counter mode of operation using Alice key. And after decrypting it will learn that the mail is supposed to be delivered to Charlie's mailbox and it will forward the mail to the inbox of the Charlie instead of the email box of the Bob and neither Alice nor mail server will be aware of this fact, right.

(Refer Slide Time: 16:38)



Active Adversary Can Break CPA-secure Schemes

$u||m$: u = To Bob@mail.com
k_A
c
$\mathrm{Enc}_{CTR}(k_A, u||m)$
$\hat{c}$
$\hat{u}||m := \mathrm{Dec}_{CTR}(k_A, \hat{c})$
$\hat{u}$ = To Charlie@mail.com
m
k_A, k_B
k_B

- ❑ Adversary has "control" over what gets decrypted --- decryption oracle (DO) service
  - ❖ By seeing decryption of a "related" ciphertext (in this case $\hat{c}$), adversary can find out what was encrypted in the original ciphertext (c)

So what we have seen here is basically, a malicious Charlie or an active Charlie can introduce a related ciphertext by exploiting the malleability property of the counter mode of operation. And it can end up identifying what was encrypted in the actual ciphertext and this violates the privacy requirement of our encryption process. I stress that this does not mean that the counter mode of operation is not secure.

It is secure under the CPA attack model where the adversary is only assumed to be an eavesdropper and it can only issue encryption oracle service. But now, the attack that we had seen here is the more powerful attack, which is not captured by the CPA attack model. Namely, the attack that we had seen in this example corresponds to the fact that adversary has control over what gets decrypted. Namely, it gets access to the decryption oracle service.

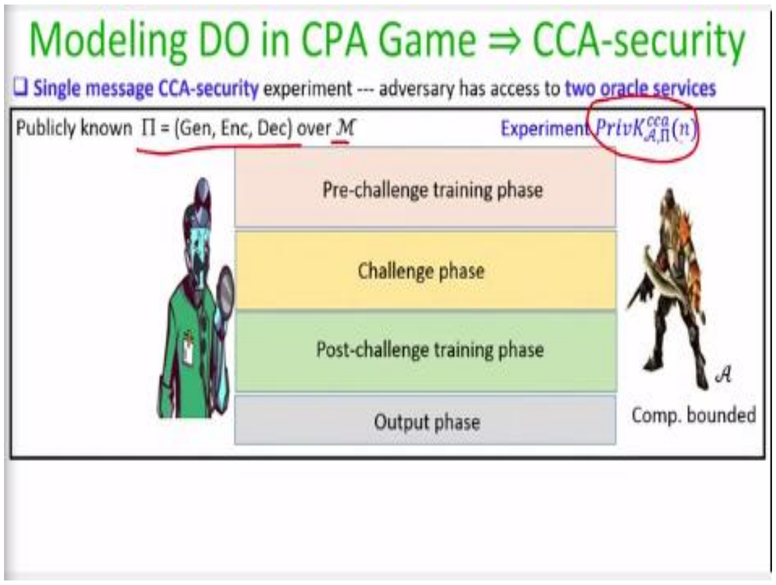
Because the malicious Charlie, what it has done is, its goal was to identify what is encrypted in c. And to do that it has prepared a modified ciphertext and forwards that ciphertext to the receiver namely the mail server, who actually decrypts the modified ciphertext oblivious of the fact that actually it is a modified ciphertext coming from an adversary and it naively decrypts the ciphertext and end up sending back the corresponding plain text to the adversary.

So that you can imagine as access to a decryption oracle service which was not the case in the CPA attack model. In the CPA attack model adversary is only restricted to getting access to encryption oracle service. We do not assume that it has access to the decryption oracle service. And the power that malicious adversary is getting by having access to the decryption oracle service is that it is basically exploiting the malleability property of your underlying cipher.

Namely, it is goal was to identify what is encrypted in c, but to do that it prepares a related ciphertext, right namely $\hat{c}$ in this case. And find out what exactly is the underlying plain text which is encrypted in $\hat{c}$. And then by knowing the relationship between the plain text which was encrypted in c, and the plain text, which is encrypted in $\hat{c}$, it can identify what was it actually encrypted in c.

So, this gives an adversary now more power compared to an adversary, who is eavesdropper and in the CPA attack model, right. So, in this specific example, we have shown the attack with respect to the counter mode of operation. And I leave it as an exercise for you that all the CPA secure mode of operations that we had discussed till now, namely the OFB mode, and CBC mode all of them can be broken, if we go to the CCA attack, if we go to an attack model, where the adversary gets access to the decryption oracle service apart from the encryption oracle service.

(Refer Slide Time: 19:42)



Modeling DO in CPA Game ⇒ CCA-security

- Single message CCA-security experiment --- adversary has access to two oracle services

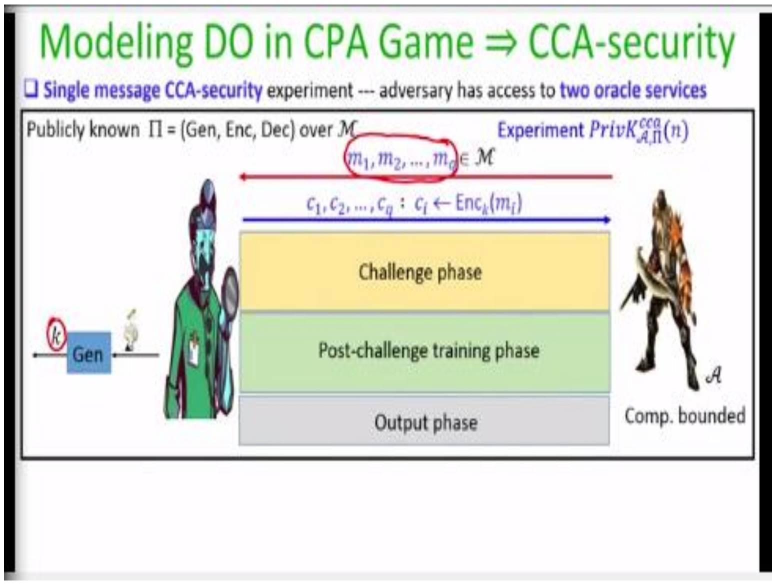
Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$ — Experiment $PrivK^{cca}_{\mathcal{A},\Pi}(n)$

So, the discussion that we had till now motivates now that we have to model the decryption oracle service in the CPA game. And that leads to a more powerful attack model or more powerful notion of security which we call us CCA security. And as we have done for the CPA world where we first defined the single message of CPA security followed by a multi message CPA security.

We are going to follow the same exercise in the CCA world as well. That means we will start with the definition of single message CCA security. And then we will proceed to the definition of multi message CCA security and so on. So the essence of single message CCA security is that adversary's goal is to distinguish between an encryption of m_0 versus an encryption of m_1 where m_0 and m_1 have chosen by the adversary itself, even if the adversary gets access to 2 oracle services, namely the encryption oracle service as well as decryption oracle service. So more precisely, we have a publicly known encryption scheme over some publicly known plaintext space. And the nomenclature of the experiment is $PrivK^{cca}_{\mathcal{A},\Pi}$ (n). We are in the CCA world, so that is why the superscript CCA and we have the security parameter n.

And the game is played between an adversary and an experiment or a hypothetical verifier where we have a pre challenge training phase, challenge phase followed by a post challenge training phase and an output phase.

(Refer Slide Time: 21:15)



Modeling DO in CPA Game ⇒ CCA-security

- Single message CCA-security experiment --- adversary has access to two oracle services

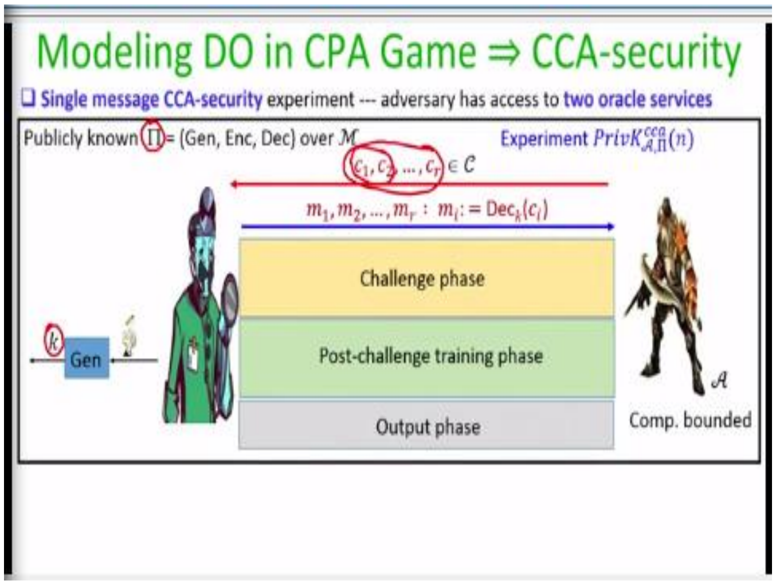
Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$
Experiment $PrivK^{cca}_{\mathcal{A},\Pi}(n)$
$m_1, m_2, \dots, m_q \in \mathcal{M}$
$c_1, c_2, \dots, c_q$: $c_i \leftarrow Enc_k(m_i)$
Challenge phase
Post-challenge training phase
Output phase
k
Gen
$\mathcal{A}$
Comp. bounded

So the pre challenge training phase is similar to the pre challenge training phase of the CPA model with some modifications. So now the adversary can get access to the encryption oracle service as well as decryption oracle service. That means it can adaptively query for the encryption of any number of messages of its choice from the plain text space as long as the number of messages are bounded by some polynomial function of the security parameter.

And to respond to this encryption oracle queries, the experiment or the challenger runs the key generation algorithm obtains the key which is unknown to the attacker. And experiment or the challenger encrypts all the messages which are queried for the encryption oracle service and the corresponding ciphertext are sent back to the adversary. So even though in this picture I have shown that adversary is querying for all the messages in the single shot.

But that may not be the case, adversary can submits its queries for the encryption oracle service adaptively. That means, it can first ask for the encryption of m_1 and based on the response it can decide what should be the m_2 that for which it should ask for the encryption oracle service and so on.

(Refer Slide Time: 22:33)



Modeling DO in CPA Game ⇒ CCA-security

❑ Single message CCA-security experiment --- adversary has access to two oracle services

Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$
Experiment $PrivK^{cca}_{\mathcal{A},\Pi}(n)$
$c_1, c_2, \ldots, c_r \in \mathcal{C}$
$m_1, m_2, \ldots, m_r : m_i := \mathsf{Dec}_k(c_i)$
Challenge phase
Post-challenge training phase
Output phase
k
Gen
$\mathcal{A}$
Comp. bounded

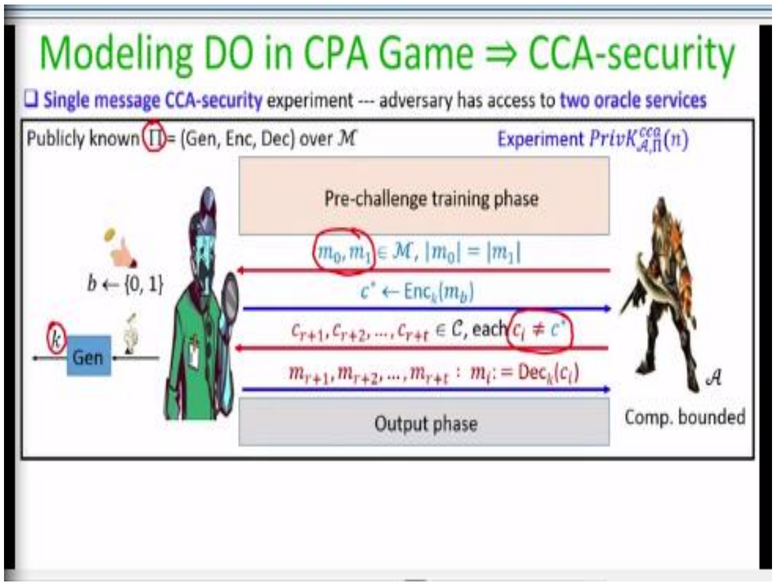
And not only the adversary can ask for the encryption oracle service, it can also ask for the decryption oracle service. Namely, the adversary since it knows the description of the encryption algorithm, decryption algorithm, plain text space and a ciphertext space, adversary will know the ciphertext space. And hence it can ask for the decryption of any number of ciphertext from the ciphertext space as long as the number of ciphertext are upper bounded by polynomial function of the security parameter.

And for responding to the decryption oracle queries, the challenger or the experiment has to decrypt all the submitted ciphertext as per the decryption process of the underlying scheme, using the same unknown key k. Again, the adversary can submit its decryption oracle queries adaptively. That means it can say for example, asked first for the decryption of an arbitrary c_1 and c_2.

And then based on the response that it sees, it can decide what should be c_3 and so on. Moreover, the adversary is allowed to overlap its encryption oracle queries and decryption oracle queries in any arbitrary order. There is no restriction that it should first ask for encryption oracle queries, submit its encryption oracle queries. And then only it should submits its decryption oracle queries.

There is absolutely no restriction, it can go in any arbitrary order as long as everything is polynomially bounded.

(Refer Slide Time: 23:56)


Modeling DO in CPA Game ⇒ CCA-security

❑ Single message CCA-security experiment --- adversary has access to two oracle services

Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$
Experiment $PrivK^{cca}_{\mathcal{A},\Pi}(n)$
Pre-challenge training phase
$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$
$b \leftarrow \{0,1\}$
$c^* \leftarrow \mathsf{Enc}_k(m_b)$
k
Gen
$c_{r+1}, c_{r+2}, \ldots, c_{r+t} \in \mathcal{C}$, each $c_i \neq c^*$
$m_{r+1}, m_{r+2}, \ldots, m_{r+t}$: $m_i := \mathsf{Dec}_k(c_i)$
$\mathcal{A}$
Output phase
Comp. bounded

And once the pre challenge training phase is over, the adversary goes to the challenge phase where it picks a pair of messages from the plain text space. But the restriction that their length should be the same. Apart from that there is absolutely no restriction, the pair of challenge plaintext which it is submitting, it could be any plain text for which it might have already asked for the encryption oracle service.

And to respond to the challenge plain text, the experiment randomly selects one of the messages with probability 1/2, it could be m_0 or with probability 1/2, it could be m_1. And once the challenge plaintext m_b is decided by the experiment, the experiment encrypts that challenge plaintext and the challenge ciphertext to c* is given to the adversary. And the challenge for the adversaries to identify what exactly is encrypted in c*.

Now, we go to the post challenge training phase where again adversary can adaptively ask for encryption of any messages of it is choice, right. It can even ask for the encryption of m_0, it can ask for the encryption of m_1 or any plaintext of it is choice. And the experiment or the challenger responds by encrypting the corresponding messages, again using the same unknown key k.

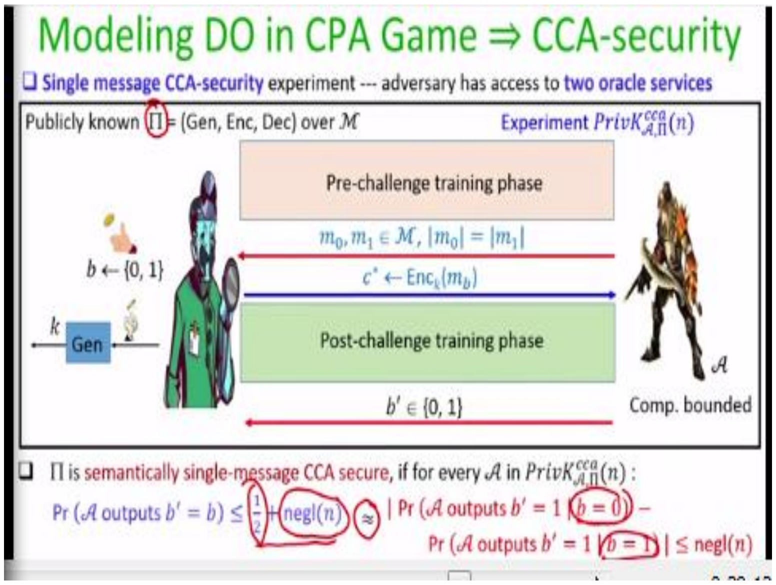
And not only that, the adversary can ask for the decryption oracle service, namely it can submit any ciphertext of its choice. With the only restriction that the ciphertext for which it is asking for the decryption oracle service, it should be different from c*, namely it should be different from the challenge ciphertext. Because, if the adversary is allowed to get the decryption oracle service even for c* then easily it can identify what exactly is encrypted in c* whether it is m_0 or m_1.

And there is no way we can define any we can give any meaningful notion of security. Also, this models the reality right. If we go back to the example that we had seen namely the email service application, right. So, the goal of the adversary or the malicious Charlie was to identify what is encrypted in c without actually getting the decryption oracle service for c. Because if the malicious Charlie can get the decryption oracle service even for c.

Then it can trivially identify what exactly is the email which Alice wants to communicate to Bob. The interesting part there was that the adversary or the malicious Charlie was given access to the decryption oracle service for any ciphertext different from the ciphertext c, which the Charlie is interested to decrypt. So to model that, in the CCA game, we put this restriction that once the challenge ciphertext is given to the adversary, adversary cannot submit a decryption oracle service for the challenge ciphertext.

Apart from that it can modify any number of bits of the challenge ciphertext and those ciphertext it can submit for decryption as some query for the description oracle service. And the experiment should respond back to those decryption oracle queries by decrypting all those ciphertext and returning back the corresponding plaintext. And finally, the adversary outputs, what plaintext is exactly encrypted in the challenge ciphertext.

(Refer Slide Time: 27:10)



Modeling DO in CPA Game ⇒ CCA-security

❑ Single message CCA-security experiment --- adversary has access to two oracle services

Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$

Experiment $PrivK^{cca}_{\mathcal{A},\Pi}(n)$

Pre-challenge training phase

$m_0, m_1 \in \mathcal{M}$, $|m_0| = |m_1|$

$b \leftarrow \{0, 1\}$

$c^* \leftarrow \text{Enc}_k(m_b)$

k Gen

Post-challenge training phase

$b' \in \{0, 1\}$

$\mathcal{A}$

Comp. bounded

❑ Π is semantically single-message CCA secure, if for every $\mathcal{A}$ in $PrivK^{cca}_{\mathcal{A},\Pi}(n)$:

$\Pr(\mathcal{A} \text{ outputs } b' = b) \leq \frac{1}{2} + \text{negl}(n)$ $\approx$ $|\Pr(\mathcal{A} \text{ outputs } b' = 1 \mid b = 0) -$
$\Pr(\mathcal{A} \text{ outputs } b' = 1 \mid b = 1)| \leq \text{negl}(n)$

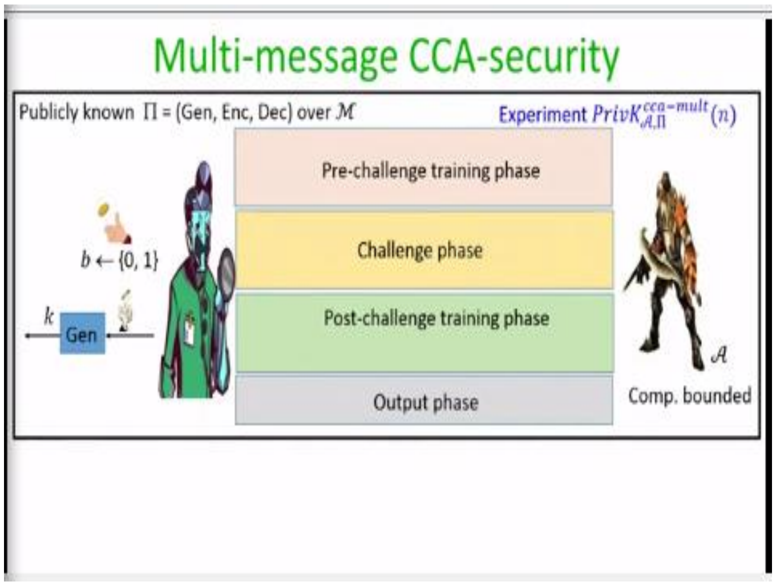
Namely it outputs a bit b’ which could be 0 or 1. Now our definition of CCA security or CCA message single message CCA security is that we say that, this publicly known encryption process Π is single message semantically secure in the CCA world or single message CCA secure, if for any polynomial time adversary A participating in this CCA game, there exist a negligible probability or negligible function, such that the probability of our adversary correctly identifying the plaintext encrypted in c* is upper bounded by ½ + negl(n), right. So we say that if adversary correctly identifies the message which is encrypted in c*, then the output of the experiment is 1. That means adversary has won the experiment otherwise we say that in the experiment adversary has lost the experiment.

So, the definition of the security is that adversary should not be able to win the experiment namely, it should not be able to output b’ = b except with probability 1/2 + negl(n), why half plus negligible, because there is a always a trivial attack, namely a guessing attack where adversary A can guess whether it is m_0 or whether it is m_1 which is encrypted in c*. And the success probability of this guessing attack is 1/2.

Apart from that we are giving an extra negligible advantage to the adversary to identify what is encrypted in c* because we are in the computationally secure world. An alternate definition for the single message CCA security is that for any polytime adversary participating in this game, the distinguishing advantage of the attacker is upper bounded by a negligible function.

Namely, it does not matter whether it is m_0 which is encrypted in c*, or whether it is m_1 which is encrypted in c*. In both the cases the response of our attacker should be the same, namely in both cases, it should output with the same output, namely say b' =1. And it can be proved formally that both these conditions are equivalent to each other. Namely, if we have an encryption process, which satisfies the first definition, then it implies that it also satisfies the second definition and vice versa. So depending upon our convenience, we can use any of these 2 definitions.

(Refer Slide Time: 29:25)

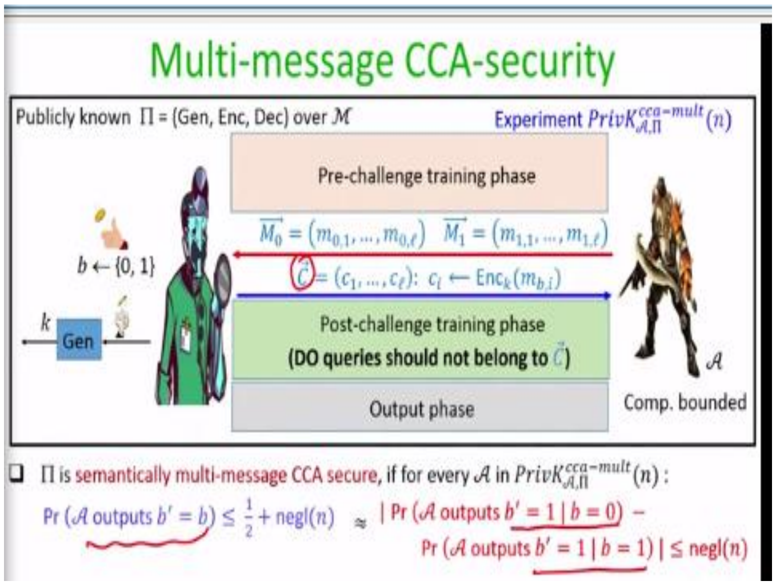


Multi-message CCA-security

Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$
Experiment $PrivK_{\mathcal{A},\Pi}^{cca-mult}(n)$
Pre-challenge training phase
Challenge phase
Post-challenge training phase
Output phase
$b \leftarrow \{0,1\}$
k
Gen
$\mathcal{A}$
Comp. bounded

So, now let us go to the multi message CCA security game, which is more or less the same as the single message CCA security game. Namely, we have a pre challenge training phase, a challenge phase, post challenge training phase and an output phase.

(Refer Slide Time: 29:40)



Multi-message CCA-security

Publicly known Π = (Gen, Enc, Dec) over $\mathcal{M}$

Experiment $PrivK_{\mathcal{A},\Pi}^{cca-mult}(n)$

Pre-challenge training phase
$\overrightarrow{M_0} = (m_{0,1}, \ldots, m_{0,\ell})$ $\overrightarrow{M_1} = (m_{1,1}, \ldots, m_{1,\ell})$
$b \leftarrow \{0, 1\}$
$\vec{C} = (c_1, \ldots, c_\ell)$: $c_i \leftarrow \mathrm{Enc}_k(m_{b,i})$
k
Gen
Post-challenge training phase
(DO queries should not belong to $\vec{C}$)
Output phase
$\mathcal{A}$
Comp. bounded

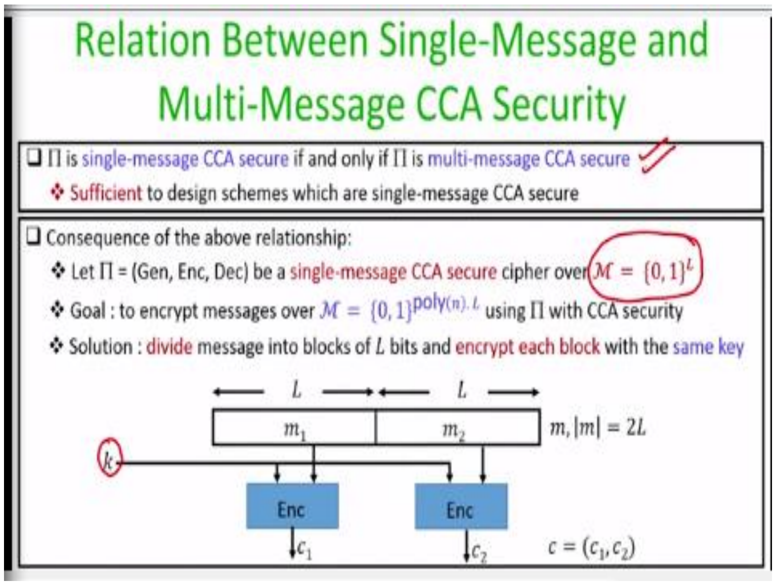
- Π is semantically multi-message CCA secure, if for every $\mathcal{A}$ in $PrivK_{\mathcal{A},\Pi}^{cca-mult}(n)$:

$\Pr(\mathcal{A}\text{ outputs } b' = b) \leq \frac{1}{2} + \mathrm{negl}(n)$ $\approx$ $|\Pr(\mathcal{A}\text{ outputs } b' = 1 \mid b = 0) -$
$\Pr(\mathcal{A}\text{ outputs } b' = 1 \mid b = 1)| \leq \mathrm{negl}(n)$

The difference is only in the challenge phase, where now the adversary can submit a pair of vector of messages. And the experiment responds by randomly choosing one of the 2 vectors with equal probability and encrypting all the messages in that corresponding vector. And in the post challenge training phase, the adversary is prevented from getting the decryption oracle service for any ciphertext in the challenge ciphertext vector. So these are the modification in the multi message CCA security game.

And our definition is we say, encryption process is multi message CCA secure, if for any polynomial time adversary the probability of adversary winning the experiment. Or correctly identifying which vector has been encrypted is upper bounded by ½ + negl(n) of the security parameter. Or equivalently the distinguishing advantage of the adversary is upper bounded by some negligible function in the security parameter. So, that is our definition of multi message CCA security.

(Refer Slide Time: 30:39)



Relation Between Single-Message and Multi-Message CCA Security

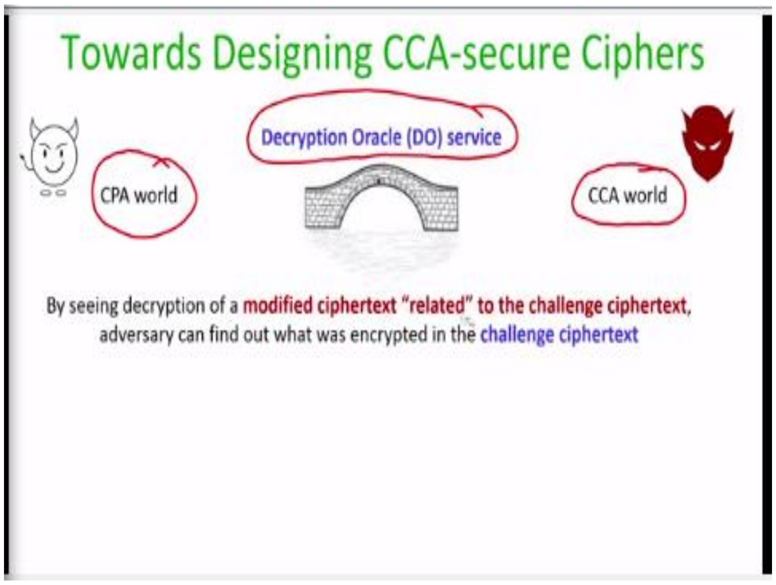
- ❑ Π is single-message CCA secure if and only if Π is multi-message CCA secure
  - ❖ Sufficient to design schemes which are single-message CCA secure
- ❑ Consequence of the above relationship:
  - ❖ Let Π = (Gen, Enc, Dec) be a single-message CCA secure cipher over $\mathcal{M} = \{0,1\}^L$
  - ❖ Goal : to encrypt messages over $\mathcal{M} = \{0,1\}^{\text{poly}(n) \cdot L}$ using Π with CCA security
  - ❖ Solution : divide message into blocks of L bits and encrypt each block with the same key

And interestingly, as it was the case in the CPA world where the single message CPA security and multi message CPA security are equivalent. We can prove formally that even in the CCA world, single message security and multi message security are equivalent, right. That means it is suffice to design encryption process which has single message CCA secure. Because this theorem gives you the guarantee that if it is single message CCA secure, then it is also multi message CCA secure.

And the consequence of this relationship between single message CCA security and multi message CCA security is that, if you have a single message CCA secure cipher for fixed length messages say for over a plaintext space consisting of all bit strings of length L. And if you want to encrypt a larger message consisting of several blocks of L bits, then it is suffice to divide your message into several blocks of L bits.

And encrypt each of the blocks of the bigger message by running an instance of the fixed length encryption process. And with the same key k, that is you can reuse the same key for encrypting each of the blocks of L bits of the larger message. And this relationship between single message CCA security and multi message CCA security gives you the guarantee that this overall process of dividing your larger message into individual blocks of L bits, and encrypting each block independently but with the same key will give you overall CCA security, right.

(Refer Slide Time: 32:10)



Towards Designing CCA-secure Ciphers

Decryption Oracle (DO) service
CPA world
CCA world

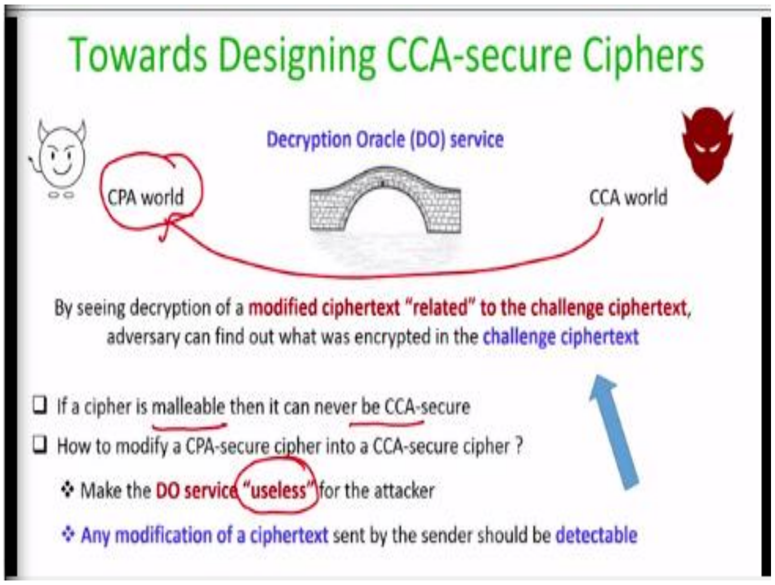
By seeing decryption of a modified ciphertext "related" to the challenge ciphertext, adversary can find out what was encrypted in the challenge ciphertext

So we now have the definition of CCA security. And our next goal will be to design candidate encryption schemes which are CCA secure, right. And what we are going to do is we will see a generic approach. Namely, we will see that how we can take encryption schemes which are secure in the CPA world. And what are the modifications or what are the additional things we need to add to those encryption algorithms to ensure that those encryption process remain secure, even if we take them to the CCA world.

So before going into that, let us try to understand the difference between the CPA world and the CCA world. The only difference is that in the CCA world adversary is given an additional access namely, it is given an additional access to the decryption oracle service. And the way this gives more power to the adversary is that, the adversary can exploit the malleability feature of the underlying encryption process.

That means what the adversary can now do is that, if it wants to identify what is encrypted in a particular challenge ciphertext. Then it can prepare a modified related ciphertext and get the decryption oracle service for the modified ciphertext. And then, by exploiting the relationship between the plaintext in the modified ciphertext and in the original ciphertext, it can identify what exactly was encrypted in the challenge ciphertext.

(Refer Slide Time: 33:38)



Towards Designing CCA-secure Ciphers

Decryption Oracle (DO) service
CPA world
CCA world

By seeing decryption of a modified ciphertext "related" to the challenge ciphertext, adversary can find out what was encrypted in the challenge ciphertext

- If a cipher is malleable then it can never be CCA-secure
- How to modify a CPA-secure cipher into a CCA-secure cipher ?
  - Make the DO service "useless" for the attacker
  - Any modification of a ciphertext sent by the sender should be detectable

And we had concretely seen that this happens when we have seen the counter mode of operation being used in the context of email service application, right. So this clearly proves or basically we can formally prove that if your encryption process is malleable, right, then it can never be CCA secure because of this fundamental fact, right. So now our goal will be if we are at all interested to design a CCA secure encryption process.

We have to ensure that how do we have to do or we have to ensure that the CPA secure cipher is modified in such a way that the decryption oracle service kind of become useless for the attacker. And what I meant by useless in this context is if there is a malicious adversary which tries to modify a ciphertext and try to get the decryption oracle service for that, that should not be possible for the adversary namely, any modification of the ciphertext should be detected either by the sender or by the receiver.

And as soon as it is detected, the receiver can simply reject those modified ciphertext. If we can somehow ensure this, then what basically this will ensure is that the decryption oracle service which the adversary was getting will become useless for the adversary. And hence from the CCA world we will back to the CPA world, right. So that is a generic approach we are going to follow for designing CCA secure encryption scheme, namely we will take any existing CPA secure schemes.

And on top of that, we will make modifications, so that the decryption oracle service becomes useless for the adversary. By ensuring that any modification of a ciphertext which has been communicated by a legitimate sender and forwarded to the receiver but blocked by a malicious adversary and changed en-route to the receiver, gets detected by the receiver. So, that will be our generic approach, in the next lecture, we will see how exactly we are going to make those modifications, so that brings me to the end of this lecture.

To summarize, in this lecture we have introduced the notion of malicious adversary or active adversary, which is a more powerful notion of adversary. Where adversary is not only allowed to eavesdrop the communication between the sender and a receiver. But the adversary is allowed to modify the ciphertext, it is allowed to insert new ciphertext, reorder ciphertext and so on.

And not only that, we also assumed that that adversary is given access to the decryption oracle service. We also discussed that how a candidate CPA secure encryption process namely the counter mode of operation can be broken, if we take it to the CCA secure world, thank you.