

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

Lecture – 33
Authenticated Encryption

(Refer Slide Time: 00:34)

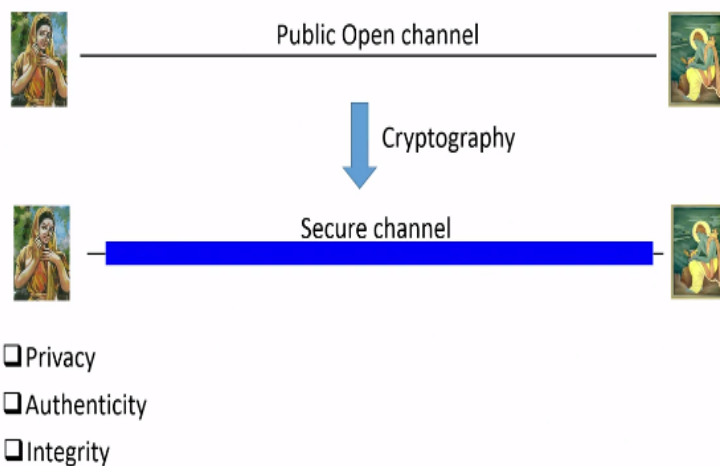
Roadmap

- ❑ Definition of Authenticated Encryption
- ❑ Implications of Authenticated Encryption

Hello, everyone, welcome to this lecture. The plan for this lecture is as follows. So, in this lecture, we will define the notion of authenticated encryption and we will see the implications of authenticated encryption. So let us begin the discussion with the goals of secure communication.

(Refer Slide Time: 00:45)

Goals of Secure Communication



So, recall that in the problem of secure communication, that scenario is the following. We have a sender and receiver with no pre-shared information whatsoever and they are connected by a public open insecure channel and our goal is basically to apply some cryptographic primitive which converts this public open channel into some virtual secure channel and what exactly I mean by virtual secure channel is that using this channel, the sender and the receiver can do communication achieving the following properties.

The first property is the privacy, namely the communication which happens over the channel it leaks no information whatsoever about the underlying messages which sender or the receiver are communicating to each other. The second property achieved here is that of authenticity, where it is guaranteed to the receiver that a bit strings which it is receiving over the channel has indeed originated from the sender and vice versa.


The third property which is achieved is that of integrity, namely if there is a malicious adversary or an active adversary, who is sitting over the channel and observing the communication and tries to modify the communication by reordering the packets or inserting new packets, right, then it will be detectable by the receiver or the sender and vice versa. So, these are the 3 goals of secure communication, and we have seen various primitives to achieve separately the privacy property and the authenticity and integrity property.

(Refer Slide Time: 02:12)

The Picture Till Now

Slide courtesy © Arpita Patra

Symmetric-key Encryption (SKE)	Message-Authentication Codes (MAC)
<ul style="list-style-type: none"> <input type="checkbox"/> Goal Privacy <input type="checkbox"/> Not necessarily provide integrity and authentication <ul style="list-style-type: none"> ❖ Easy for an attacker to produce a new valid ciphertext ❖ Easy for an attacker to manipulate an existing ciphertext and go undetected 	<ul style="list-style-type: none"> <input type="checkbox"/> Goal Integrity & Authentication <input type="checkbox"/> Not necessarily provide message privacy <ul style="list-style-type: none"> ❖ Easy for an attacker to distinguish between MAC-tags of two known messages



☐ Privacy ✓
☐ Authenticity ✓
☐ Integrity ✓

Authenticated Encryption

Namely the picture till now is as follows. So, we have extensively discussed 2 cryptographic primitives, namely symmetric encryption or SKE and message authentication codes or MAC. So, property wise, the goal of secure symmetric encryption is to achieve the privacy property

where the goal of message authentication code is to solve the integrity problem and authentication problem. If you consider the symmetric encryption, then it just solves the privacy problem.

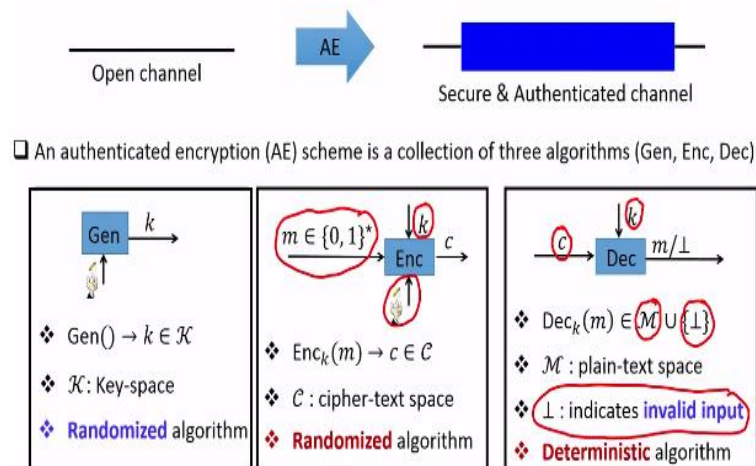
It does not solve the integrity and authentication problem and it is easy to see that whatever encryption mechanisms that we have discussed till now, for all those encryption mechanisms, it is very easy for an attacker to produce a new valid ciphertext because the ciphertext is nothing but a bit string and since the adversary will be aware of the ciphertext space, it can just produce some bit stream belonging to that ciphertext space and can forward it to the receiver and receiver will have no mechanism to verify whether the so called bit strings which it is receiving has indeed originated from the designated sender or not.

Also, it is easy for an attacker to manipulate an existing ciphertext and go undetected. That means, the symmetric key encryption that we have discussed till now, it just allows or helps us to solve the problem of privacy. Whereas if you see the message authentication codes, then the goal of the message authentication codes is to just solve the problem of integrity and authenticity and it does not provide you message privacy and it is easy to see that whatever message authentication code that we have discussed till now, for all those message authentication codes, it is very easy for an attacker to distinguish apart message authentication tags for 2 messages of adversary's choice. So that means, we cannot say that adversary cannot distinguish between the tag of message 0 versus the tag of message 1. It is very easy for the adversary to do that, right. So now, you can see that the property wise, the requirement or the goals achieved by symmetric encryption and the message authentication codes are complementary to each other, whereas our overall goal is to achieve privacy, integrity, and authenticity; all the 3 properties by a single primitive

So what we can hope for is that we can hope that or we can imagine that we should somehow combine these 2 primitives and hope that whatever comes out as a combination of these two primitives, which we call as authenticated encryption satisfies all the 3 properties, namely it gives us privacy, authenticity, and integrity. So that is what we are going to do now. We will see how to combine symmetric encryption and message authentication code to achieve or design a more powerful cryptographic primitive which we call as authenticated encryption.

(Refer Slide Time: 05:05)

Authenticated Encryption: Formal Definition



So on a very high level as I said, the goal of authenticated encryption is to take an open channel with which a sender and a receiver are connected and to give the effect of a secure and authenticated channel. So formally an authenticated encryption scheme is a symmetric encryption process and it will be a collection of 3 algorithms, namely a key generation algorithm, an encryption algorithm, and a decryption algorithm.

So syntax wise, the syntax of the key generation algorithm is that it will have no external input and it will have implicit randomness and it will produce a key from a key space and preferably it has to be a randomized algorithm because if it is deterministic algorithm, then the adversary also will be knowing what exactly is the key which sender and receiver are going to obtain by running the key generation algorithm. So that is the syntax of key generation algorithm.

The encryption algorithm takes the plain text which you want to encrypt and the key with which you want to perform the encryption and it has to be a randomized algorithm. So, it has implicit randomness and it gives you a cipher text c belonging to the ciphertext space. On the other hand, the decryption algorithm takes the ciphertext which you want to decrypt and the key with which you want to perform the decryption, and it can have 2 possible outcomes. So that is the difference now that we are encountering here when we are discussing the decryption algorithm for the authenticated encryption schemes.

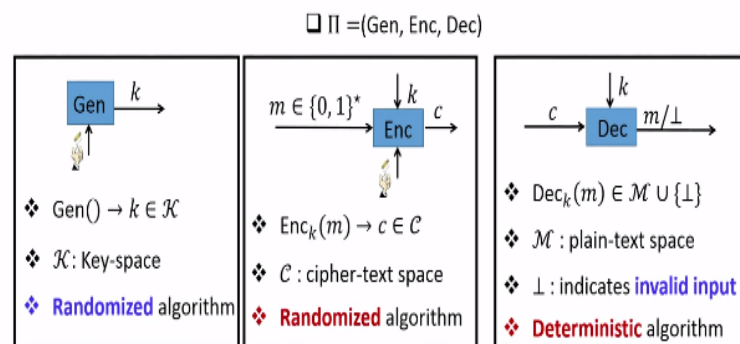
So it was till now the decryption algorithm for the primitive symmetric encryption scheme that we have discussed it has only one possible output, namely the plain text which you

obtain by decrypting the ciphertext, but when we come to authenticated encryption scheme, the output of the decryption algorithm could either belong to the plain-text space or it could be a special symbol which we called as bot, which means an invalid input and the decryption algorithm has to be a deterministic algorithm to ensure that the decryption is unambiguous.

So now, you might be wondering what exactly is the interpretation of invalid input? It will be clear very soon what exactly we mean by an invalid input or an invalid ciphertext in the context of the decryption algorithm of an authenticated encryption scheme.

(Refer Slide Time: 07:31)

Authenticated Encryption: Formal Definition



Π is an authenticated encryption (AE) scheme, if it satisfies the following:

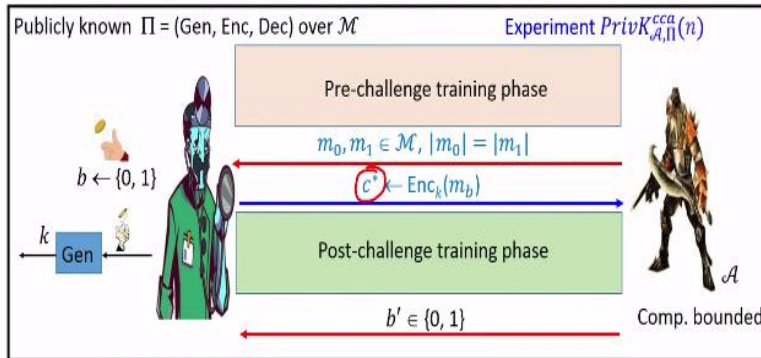
- \diamond CCA-security ✓✓
- \diamond Ciphertext integrity ✓✓

So that is the syntax of authenticated encryption scheme, and formally we say that encryption scheme as an authenticated encryption scheme if it satisfies the following 2 properties. The first property is that it should satisfy the notion of CCA-security and the second property is that it should satisfy the notion of ciphertext integrity. So now let us individually discuss each of these two properties. So we had already discussed what exactly CCA-security means, but let us again recall the definition of CCA-security.

(Refer Slide Time: 08:01)

CCA-security

□ Semantic security, even in the presence of encryption and decryption oracle



□ Π is **semantically CCA secure**, if for every \mathcal{A} in $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cca}}(n)$:

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

Basically, the goal of CCA-security is that we want to achieve semantic security namely indistinguishable encryptions even in the presence of an adversary who has access to both the encryption oracle as well as decryption oracle, right. So that makes this attack model or this notion of security more powerful compared to CPA security. In CPA security, the goal was to achieve semantic security only if the adversary has access to the encryption oracle service.

But now we want the adversary should not be able to distinguish apart an encryption of m_0 from an encryption of m_1 even if the adversary has an additional access to the decryption oracle service. So formally, this is modeled by an experiment and we have 4 phases in the experiment. We have a pre-challenge training phase, we have a challenge phase, we have a post-challenge training phase and we have an output phase. The game is played between a computationally bound adversary and an experiment.

So in the pre-challenge training phase, the adversary can adaptively ask queries from the encryption oracles. That means, it can adaptively submit plain text of its choice from the message space and in response that challenger or the experiment has to encrypt all those messages by running a key generation algorithm and obtaining a key which is not known to the attacker and all the messages for which the adversary has asked the encryption oracle service, the challenger encrypts those messages and a corresponding ciphertext are communicated back to the adversary.

Moreover, the adversary also has access to the decryption oracle service. It can ask for the decryption of any ciphertext on the ciphertext space of its choice and whenever adversary

submits decryption oracle queries, the challenger or the experiment has to decrypt all those ciphertext under the same unknown key k . So, I stress here that the same unknown key is going to be retained throughout the experiment for responding to the encryption oracle service, for responding to the decryption oracle Service, and even for preparing the challenge ciphertext.

Also, there is no restriction on the adversary on the order in which it can ask the encryption oracle queries and a decryption oracle queries. It can adaptively submit queries in any arbitrary order, right. So once the pre-challenge training phase is over, then we have the challenge phase where adversary submits a pair of challenge plain text with the only restriction being that their length should be same. To prepare the challenge ciphertext, the challenger randomly decides one of those 2 messages with probability $1/2$.

It could be the zeroth message or the first message and once it decides which message to encrypt, the challenger encrypts that challenge plain text and the challenge ciphertext c^* is communicated to the adversary and the challenge for the adversary is to distinguish apart whether c^* is an encryption of m_0 or whether c^* is an encryption of m_1 . Now in this attack, in the CCA experiment, we give the adversary additional power.

Namely once the adversary sees the challenge ciphertext, we also give the adversary access to the post-challenge training phase where it can again ask for encryptions of several messages of its choice, possibly including the challenge plain text m_0, m_1 right. So it can ask for the encryption of any number of messages of his choice and the challenger has to respond back to those queries by encrypting those messages as per the key, and the adversary can also ask for decryption oracle service.

Namely, it can submit any ciphertext of his choice for decryption with the restriction that none of this post-challenge decryption oracle queries can be same as c^* because if we do not put this restriction, then trivially the adversary can win the game by asking for the description of c^* and once it sees the description of c^* , it can clearly identify whether it has seen an encryption of m_0 or m_1 , and hence we cannot define any meaningful notion of security.

So that is why in the post-challenge training phase, we prevent the adversary from making decryption oracle query for c^* . Apart from that, he can modify any number of bits of c^* and

ask for the decryption of those ciphertext, and in response, the challenger of the experiment has to decrypt those ciphertext and return back the corresponding plain text to the adversary. Once the adversary is properly trained with respect to the pre-challenge queries, post-challenge queries, the adversary has to now decide whether it has seen an encryption of m_0 or whether it has seen an encryption of m_1 .

So it basically outputs a bit b' . The security definition is we say that the encryption process Π is semantically CCA secure. If every poly-time adversary participating in this experiment, there exists some negligible function such that the probability that adversary wins the experiment or ensures that $b' = 1$ is upper bounded by $1/2$ plus that negligible function. Equivalently, the other way to interpret this definition is that it does not matter whether its message m_0 which is encrypted in c^* or whether its message m_1 which is encrypted in c^* , in both the cases, the response of the adversary should be almost identical, say $b' = 1$ except with some negligible probability. So that is the other definition of CCA-security and we can prove that if we have a scheme which satisfies the first condition, then it implies that it also satisfies the second condition and vice versa. So depending upon our convenience, we can use any of these 2 conditions to prove or disprove the CCA-security of a given encryption process. So, we had defined what exactly we mean by CCA-security.

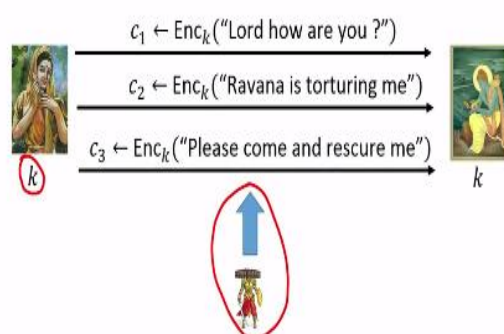
So, now let us see what exactly we mean by ciphertext integrity. So that is a new property, which we also require from an authenticated encryption scheme.

(Refer Slide Time: 14:05)

Ciphertext Integrity (CI)

□ If a cipher has ciphertext integrity then an adversary should not be able to do the following

$\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$

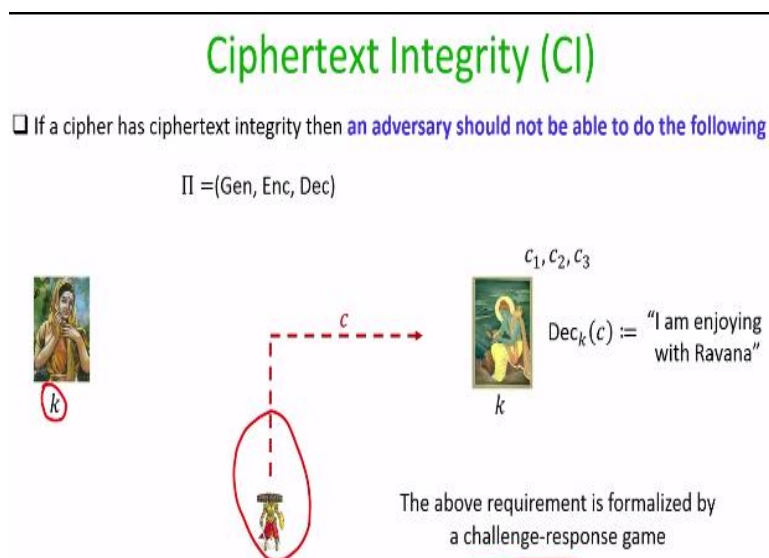


So what exactly is ciphertext integrity? So, informally if an encryption process has ciphertext integrity, then that means that an adversary should not be able to do the following. So imagine we have an encryption process and there is a pre-shared key generated as per the key generation algorithm and available between the sender and the receiver and not known to anyone, and imagine that sender has encrypted several legitimate plain text.

Say the sender has encrypted the message “Lord how are you” under the key k which is not known to the attacker and say she again encrypts the message “Ravana is torturing me” and communicates the ciphertext, and again she has encrypted the message “Please come and rescue me” and the ciphertext is communicated over the channel and imagine that there is a malicious adversary or an active adversary, right, who has eavesdropped all this encrypted communication.

So, the adversary here does not know what exactly are the underlying messages, right, which have been encrypted and it also does not know the value of the key, but it has access only to the legitimate ciphertext which have been computed and communicated by the sender to the receiver.

(Refer Slide Time: 15:16)



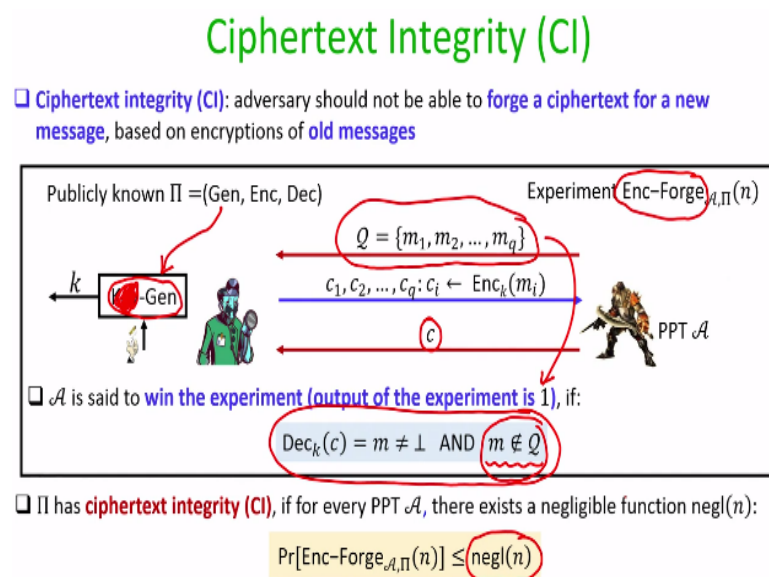
Now, if we say that our encryption processes has ciphertext integrity, then it should not be possible for this malicious adversary to now inject a new ciphertext or a bit string say c , which was not communicated by the sender, but now what this adversary is trying to do is this adversary is trying to send this ciphertext on the behalf of the sender such that the

ciphertext when it gets decrypted at the receiving end, it corresponds to a plain text which was never communicated by the sender.

So the goal of the ciphertext integrity is to prevent an adversary from doing that. Namely, it should not be possible for an adversary to see several legitimate ciphertext from the past sessions and based on that even without knowing the key, it should not be possible for the attacker to come up with a new ciphertext on the behalf of the sender and communicate it to the receiver such that that in a new ciphertext gets decrypted legitimately at the receiving end.

So, that is what we want to prevent to ensure that our encryption process has ciphertext integrity and this requirement that I have demonstrated here pictorially can be formalized via challenge-response game.

(Refer Slide Time: 16:31)



So let us see the ciphertext integrity experiment. So informally, the goal here is that the adversary should not be able to forge a ciphertext for a new message based on the encryptions of old messages. So we model this requirement by this experiment, which we call as Enc – Forge_{A,Π}(n) played between a computationally bounded adversary and a verifier or an experiment and we have a training phase here and we have an output phase here.

So in the training phase, we give the adversary a chance to get trained itself, namely we allow the adversary to ask for encryptions of several messages of its choice as per the encryption algorithm of the scheme Π. So it adaptively submits a set of encryption queries, m_1, m_2, m_q and to respond to these queries, experiment runs the key generation algorithm and obtains a

uniformly random key. So sorry for the typo here, this Key-Gen actually should be just algorithm Gen because I am denoting the key generation algorithm of the scheme by Π as Gen. So sorry for the typo, it should not be key and it should be just Gen. So as per the key generation algorithm, the experiment obtains a key and it responds to the query that had been submitted by the adversary by encrypting all those messages under this key as per the encryption algorithm. Now the challenge for the adversary is to forge a ciphertext or a new ciphertext. So, basically the goal of the adversary is to come up with a ciphertext c and the rules of the experiment is as follows.

We say that the adversary has won the experiment or the output of the experiment is 1 if the following 2 conditions hold. The first condition is that the decryption of the ciphertext c should be some value from the plain text space and not the special symbol \bot and not only that on decryption whatever message we obtain, it should not belong to the set of queries for which the adversary has already seen the ciphertext. So, what exactly we mean here is that what exactly we are trying to model to this experiment is to remember the scenario that we have discussed between the sender and the receiver.

There the goal of the adversary was to see several ciphertext which have been legitimately computed and communicated by the sender to the receiver, and based on that the goal of the adversary was to come up with a new ciphertext c which when decrypted gives you a plain text which was never communicated and encrypted by the sender and sent to the receiver that exactly is what we are trying to capture through this experiment. So, this query is for which the adversary has got the encryption.

It corresponds to the previous inscriptions or the ciphertext which the adversary has already seen, and based on that, the adversary's goal is to come up with a forgery, namely a new ciphertext, which when decrypted gives you a plain text which does not belong to the set of queries Q . So now, you might be wondering that why we are putting this restriction that decrypted plain text that we obtain after decrypting this c should not belong to this set of queries Q . If we do not put this restriction, then there is a very simple strategy for the adversary to win the game.

Namely, it can set c to be any of the c_i values that he has seen as a response to the queries that he has asked from the experiment and that can be considered as some kind of replay attack

where the adversary is simply replaying or just inserting a ciphertext which have been already legitimately communicated by the sender to the receiver, but replay attack is not what we want to prevent through the ciphertext integrity property. These replay attacks, they are taken care by through different mechanisms which are we are not going to discuss here.

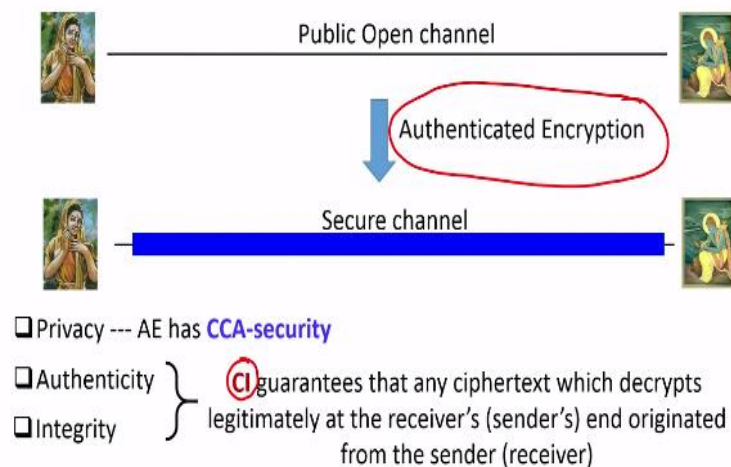
The goal of the ciphertext integrity is to prevent an adversary from forging a new ciphertext, which was never communicated by the sender to the receiver and that is why the required definition or the way we are defining that the adversary has won the game is as follows. First of all, the forged ciphertext should give you back a legitimate output and not only that legitimate output should be different from the set of queries for which the adversary has got the encryption oracle service.

Now, the formal definition is we say that an encryption process has ciphertext integrity property if for every poly-time adversary participating in this experiment, the probability that it can forge a new ciphertext is upper bounded by some negligible function. I stress that we cannot put a condition that in this definition the probability of forgery should be 0 because there is always a guessing strategy by the adversary where it can just guess a value of candidate c .

It may turn out that with nonzero probability that the guessed c indeed is a legitimate ciphertext and it decrypts to a plain text m which is not belonging to the set of queries Q , but what we want from an encryption process is that is the best an adversary can do and if we achieve that, then we say that our encryption process has ciphertext integrity property.

(Refer Slide Time: 21:43)

AE Implies Secure Communication



So, now let us see what exactly are the consequences of an authenticated encryption scheme? So remember our goal is to convert a public open channel between a sender and a receiver into a virtual secure channel and your authenticated encryption exactly achieves that. Namely, if the sender encrypts its plain text using an authenticated encryption scheme, then it gives her the effect as if that she is talking to the receiver over a virtual secure channel.

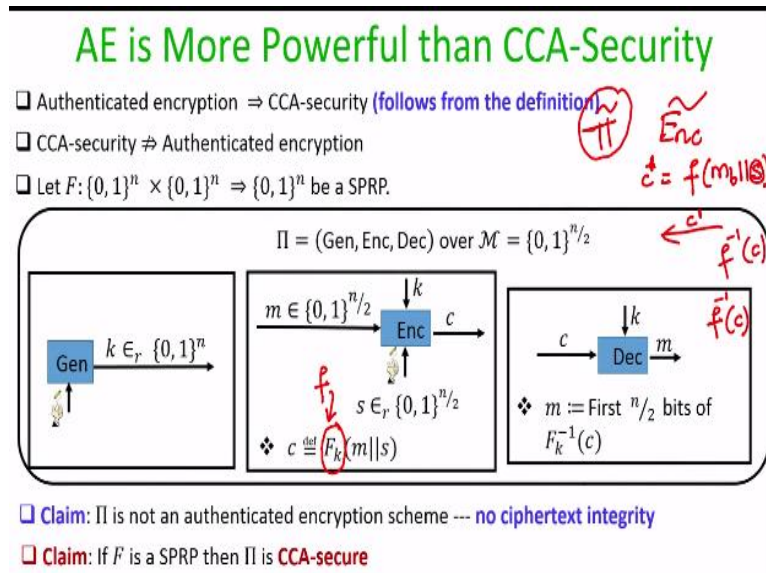
Why that is so because the privacy property or the privacy of a communication is achieved because of the fact that your authenticated encryption has CCA-security and the authenticity and the integrity of our communication are ensured by the ciphertext integrity property. This is because if at all any ciphertext is decrypted legitimately at the receiving end, then the receiver is sure that with very high probability it has indeed originated from the designated sender because the receiver is going to decrypt the ciphertext using the key k and only a designated sender has the same key k .

So if at all, ciphertext received by the receiver has been decrypted legitimately and does not give output bot, then from the ciphertext integrity property, it follows that indeed that ciphertext was not inserted by an adversary. It has indeed been communicated by the sender and it has not been tampered upon and the same guarantee is given even if a receiver is encrypting some message as per the authenticated encryption scheme using the key shared with the sender.

If the ciphertext decrypts back legitimately at the sender's end, then it gives the sender the guarantee that indeed ciphertext has come from a receiver who has the same key k with

which the sender has decrypted the ciphertext. So now you can see that indeed authenticated encryption is the right notion of security if you want to achieve all the 3 properties together, namely privacy, authenticity, and integrity.

(Refer Slide Time: 23:48)



So now let us see the relationship between authenticated encryption and CCA-security. It turns out that authenticated encryption is a more powerful notion than CCA-security. Namely, it is easy to see that authenticated encryption trivially implies CCA-security because it simply follows from your definition of authenticated encryption. When we say that our encryption process is authenticated encryption, then one of the requirements is that it should also be CCA secure.

So this implication is trivial, but it turns out that CCA-security need not imply authenticated encryption. That means, what we are now going to do is we are going to give you a construction or an encryption process which is CCA secured, but it is not authenticated encryption. So, the construction is as follows. So, imagine we are given a keyed strong pseudorandom permutation and for simplicity, I am assuming that the key size, block size, and output size are all n bits where n is the security parameter.

We are going to construct an encryption process for encrypting messages of $n/2$ bits. So the key generation algorithm for this encryption process outputs a key for our strong pseudorandom permutation uniformly randomly, which will be shared between the sender and the receiver by some magical mechanism and if sender now wants to encrypt the message m of size $n/2$ bits, what it does is it internally picks a uniform randomness of size $n/2$ bits,

which I denote by s and to encrypt the message m what it does is it concatenates m with the randomness s .

So that will be now a whole block of n bits and to encrypt that block, what the sender does is it evaluates strong pseudorandom permutation with the key k and the block input being m concatenated with s and the resultant output is the ciphertext. So that is a very simple way of encrypting the message. To decrypt the ciphertext what the receiver is going to do is if the receiver receives a ciphertext c and it has the same key k , it will first compute the inverse of the value c with respect to the key k .

So that means it will compute F_k^{-1} with the input c , and on inverting it will obtain a chunk of n bits and it knows that as for the syntax of the encryption process, the last $n/2$ portions of the output that it has obtained is the randomness part, so it can simply throw it off, and it can take the first $n/2$ bits of the recovered n bits and consider it as the output plaintext. So that is your corresponding decryption process. Now, let us see whether this encryption process is CCA secure or not and whether it is an authenticated encryption scheme or not.

It turns out that this encryption process is not an authenticated encryption scheme because it does not have the ciphertext integrity property. So remember, as per the definition of authenticated encryption scheme, the scheme has to satisfy both the CCA-security property as well as the ciphertext integrity property, but the candidate encryption scheme that we have given here, it does not have the ciphertext integrity property. This is because it is very simple for an adversary to come up with a bit string, c' , which is of size n bits and that is a legitimate ciphertext as per the syntax of this encryption process.

There is no checking which we are performing at the decryption end to verify whether the received ciphertext has indeed originated from the designated sender or not. Whatever n bit chunk comes as the ciphertext, the decryption algorithm is simply going to perform or compute an inverse of that ciphertext with respect to the pseudorandom permutation and take the first $n/2$ bits of the recovered block, right. That is the way we are going to perform the decryption here.

So, it is very easy for an adversary to just cook up any ciphertext from the ciphertext space and send it to the receiver on the behalf of the sender and that simply violates the ciphertext

integrity property. Interestingly, we can prove that this encryption scheme has the CCA-security property, that means it is indeed CCA secure and I am leaving the proof of this claim as an exercise for you. The idea behind the proof of this claim is as follows. If we consider an alternate encryption scheme say $\widetilde{\text{Enc}}$, where all the instances of this keyed pseudorandom permutation are replaced by a truly random permutation.

So imagine both sender and receiver has access to a truly random permutation and to compute an encryption of $n/2$ bits, you pick a randomness of $n/2$ bits and evaluate this truly random function on the concatenated message and the randomness and analogously you perform the decryption operation to decrypt the ciphertext, you compute the inversion of the ciphertext as per the truly random permutation and just take the first $n/2$ bits of the recovered output as the plaintext.

So if we consider that to be our alternate encryption process, then we can prove that that alternate encryption process is indeed CCA secure. This is because the value of the randomness which we are using to prepare the challenge ciphertext in an instance of the CCA game will not be known to the adversary, right. So if we take this alternate encryption process in and play an instance of the CCA game and imagine that c^* is the encryption of the plaintext m_b , then the encryption of the challenge plaintext m_b will be as per this process, right.

The experiment would have picked a randomness of size $n/2$ bits concatenated with the message m_b and it would have evaluated the truly random function and that would be the challenge ciphertext. Now until and unless the adversary does not know the value of this randomness s which has been used to prepare the challenged ciphertext, adversary cannot find out whether it is seeing an encryption of m_0 or whether it is seeing an encryption of m_1 , and more importantly, this holds even if the adversary has got access to the decryption oracle query.

Namely through the decryption oracle query, say for example adversary asked for the decryption oracle service for a ciphertext c' , then through the decryption of c' , basically adversary is learning the value of the inverse of the truly random function on c , but it is very unlikely and since as per the rule of the CCA game, adversary is not allowed to ask for the

description of c' being equal to c^* , again through the description oracle queries, adversary is not going to learn the value of the randomness s which has been used to prepare the challenge ciphertext c^* .

So that means, we can prove that with very high probability this alternate scheme $\tilde{\Pi}$ will be CCA secure. Now, if we come back to the actual scheme Π , the only difference here is that we are actually replacing the truly random function or a truly random permutation by a keyed strong pseudorandom permutation, and as per the security definition of strong pseudorandom permutation, its behavior is computationally indistinguishable from the behavior of a truly random permutation.

So that means, this scheme Π also should be CCA secure. If not, then we know how to construct an adversary who can distinguish apart the behavior of a truly random permutation from the behavior of a keyed pseudorandom permutation, but that will contradict our assumption that the function F_k is a strong pseudorandom permutation. So, I am leaving those formal reductions and the details as an exercise for you that is straightforward and you should be able to do that.

So that brings me to the end of this lecture. Just to summarize, in this lecture we have introduced the notion of authenticated encryption and that is our ultimate or the gold standard for the symmetric encryption scheme because if we have an authenticated encryption scheme, then using this authenticated encryption scheme, sender and receiver can convert a publicly open channel into a virtual secure channel and perform secure communication which will achieve all the 3 goals of secure communication, namely privacy, integrity, and authentication. Thank you.