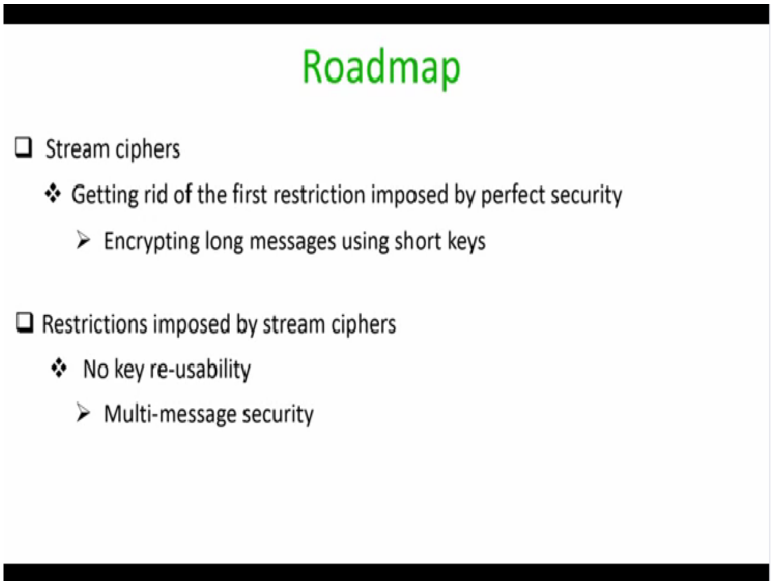
Foundations of Cryptography
Prof. Dr. Ashish Choudhury
(Former) Infosys Foundation Career Development Chair Professor
Indian Institute of Technology-Bangalore

Lecture-10
Stream Ciphers

Hello everyone, welcome to lecture 9. The plan for this lecture is as follows.

(Refer Slide Time: 00:33)

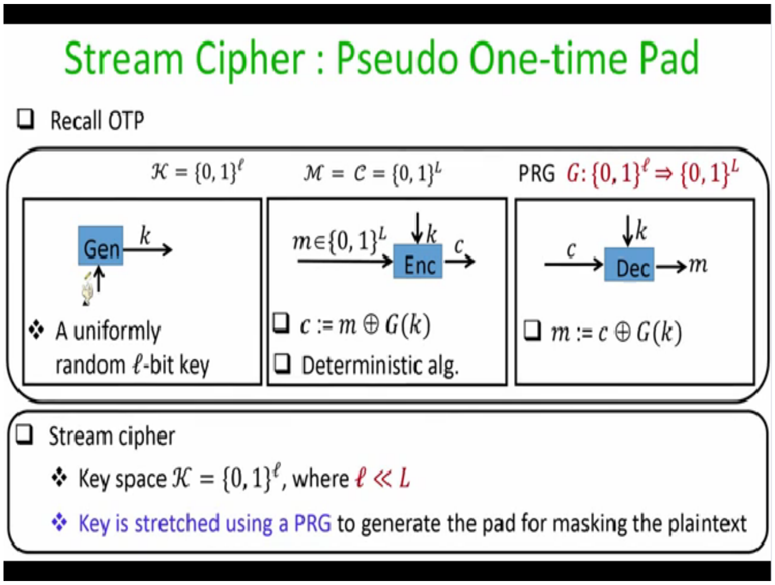


Roadmap

- ❑ Stream ciphers
  - ❖ Getting rid of the first restriction imposed by perfect security
    - ➢ Encrypting long messages using short keys
- ❑ Restrictions imposed by stream ciphers
  - ❖ No key re-usability
    - ➢ Multi-message security

In this lecture we will consider the notion of stream cipher, where we will see how to encrypt long messages using short keys with the help of pseudo random generators and we will also see the restrictions imposed by stream ciphers, namely we will see that stream ciphers does not supports key reusability. For this, we will also introduce the notion of multi message security.

(Refer Slide Time: 00:54)



Stream Cipher : Pseudo One-time Pad

❑ Recall OTP

$\mathcal{K} = \{0,1\}^\ell$ $\mathcal{M} = \mathcal{C} = \{0,1\}^L$ PRG $G: \{0,1\}^\ell \Rightarrow \{0,1\}^L$

❖ A uniformly random ℓ-bit key

❑ $c := m \oplus G(k)$

❑ Deterministic alg.

❑ $m := c \oplus G(k)$

❑ Stream cipher

❖ Key space $\mathcal{K} = \{0,1\}^\ell$, where $\ell \ll L$

❖ Key is stretched using a PRG to generate the pad for masking the plaintext

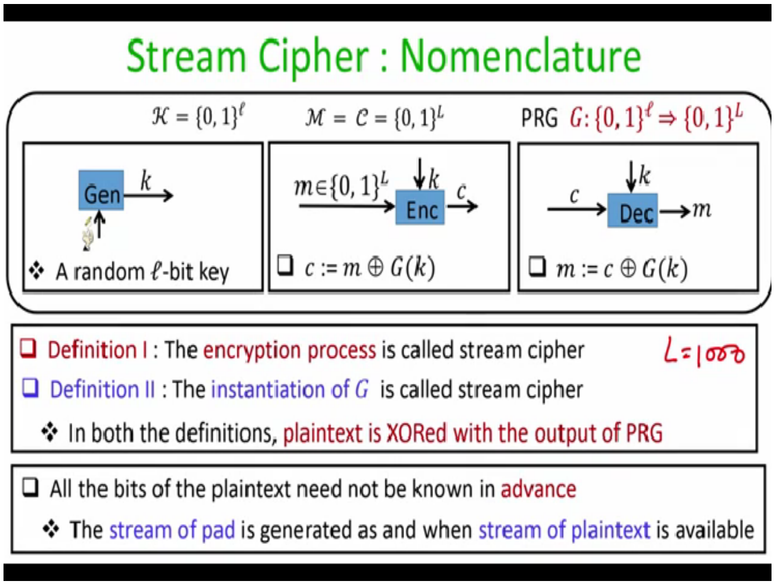
So, on a very high-level stream cipher you can imagine it is a pseudo one-time pad. So just quickly recall the one-time pad scheme where the message space, key space and ciphertext space $\mathcal{M}= \mathcal{C}= \mathcal{K} =\{0, 1\}^L$, the key generation algorithm outputs a uniformly random key of size L bits. To encrypt $m \in \{0, 1\}^L$, $c := m \oplus k$. The decryption operation is m = $c \oplus k$.

Now in the stream cipher, the key space $\mathcal{K}= \{0, 1\}^\ell$ and $\ell \ll L$. So, we have now 2 different spaces here, the key space is different and the message space and the ciphertext space are different. The message space and the ciphertext space $\mathcal{M}= \mathcal{C}= \{0, 1\}^L >> \ell$.

So, the first change that we are going to make in the stream cipher compared to the OTP is that the key generation algorithm is now going to output a key which will be of size *l* bits instead of L bits. And to perform the encryption operation, we will XOR the message with the output of a PRG invoked on the key generated by the key generation algorithm.

In the stream cipher, we assume that we also have a secure PRG extending or stretching *l* bits to an output of L bits. So, by running the pseudo random generator on the key k, we generate the mask and that mask is XORed with the message and that produces the ciphertext. The same operation we perform at the decryption.

(Refer Slide Time: 02:41)



Stream Cipher : Nomenclature

$\mathcal{K} = \{0,1\}^{\ell}$ $\mathcal{M} = \mathcal{C} = \{0,1\}^{L}$ PRG $G: \{0,1\}^{\ell} \Rightarrow \{0,1\}^{L}$

❖ A random ℓ-bit key

❑ $c := m \oplus G(k)$

❑ $m := c \oplus G(k)$

❑ Definition I : The encryption process is called stream cipher $L=1000$

❑ Definition II : The instantiation of G is called stream cipher

❖ In both the definitions, plaintext is XORed with the output of PRG

❑ All the bits of the plaintext need not be known in advance

❖ The stream of pad is generated as and when stream of plaintext is available

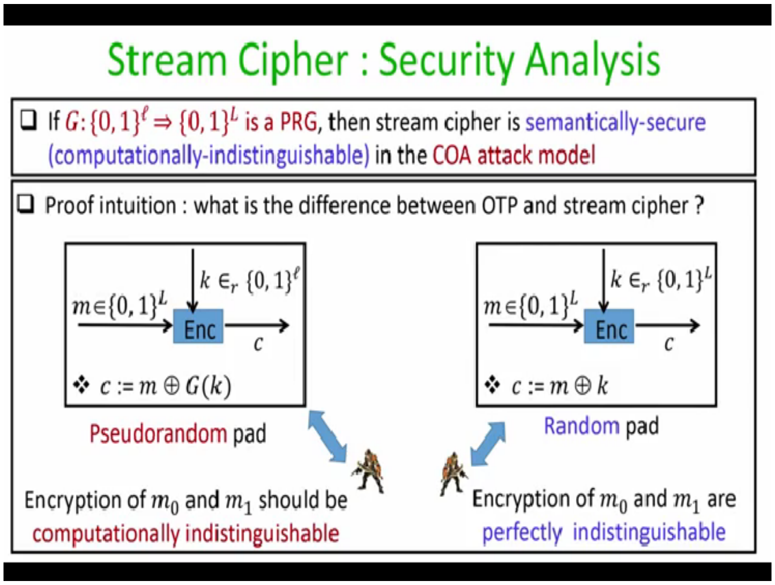
So, in stream cipher we have different nomenclature followed in the literature. We have few sources which says that the entire encryption process is called a stream cipher whereas we have other sources which says that the instantiation of the pseudo random generator is called a stream cipher. Irrespective of whether we are following definition 1 or definition 2, the important thing is that the message m is XORed with outcome of the pseudo random generator on a uniformly random seed, which is much smaller size compared to the message size.

Now, the reason that this whole encryption process or this whole system is called a stream cipher is that the bits of the plain text, all the bits of the plain text need not be known in advance that means, for the moment imagine that your L is 1000. That means, you want to design an encryption process which can encrypt messages of length 1000 bits. And for instance, imagine that you have got a message, you have obtained only say, the first 100 bits of your message.

So, since you have only the first 100 bits of the message, what you can generate is, you can run the algorithm G and produces the first 100 outputs bits of your algorithm G on the input k and that constitutes a stream with which you can XOR the first 100 bits of the message and send it to the receiving end. Next, if you receive the next few bits of the message, you can produce the next few output bits of the algorithm G on the same input K to produce the next sequence of pads with which you can XOR the next sequence of bits of the message and so on.

You can imagine that actually your algorithm G is producing the sequence or a stream of pads depending upon the sequence of bits of the message which are available to you and that is why it is called as stream cipher the whole message may not be available to you in advance.

(Refer Slide Time: 04:34)



Stream Cipher : Security Analysis

❑ If $G: \{0,1\}^{\ell} \Rightarrow \{0,1\}^{L}$ is a PRG, then stream cipher is semantically-secure (computationally-indistinguishable) in the COA attack model

❑ Proof intuition : what is the difference between OTP and stream cipher ?

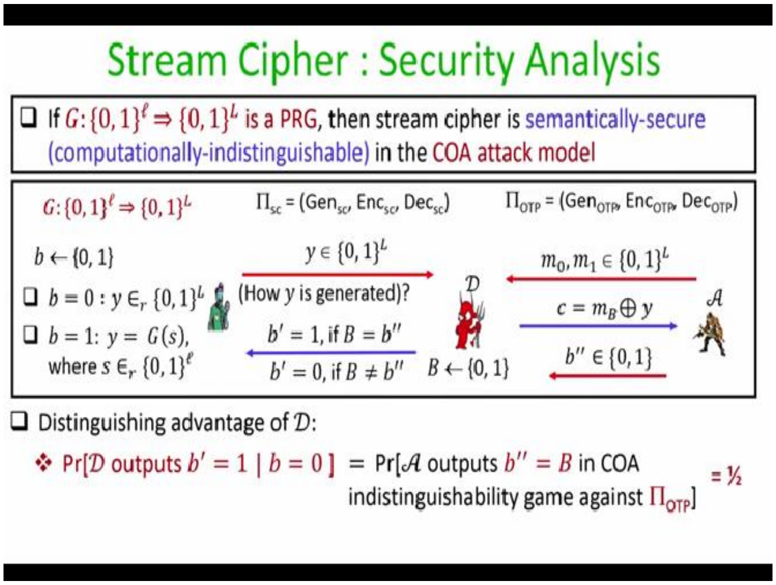
Encryption of m_0 and m_1 should be computationally indistinguishable

Encryption of m_0 and m_1 are perfectly indistinguishable

Now, let us analyze that why this stream cipher is secure. So, the claim that we want to make here is that if your algorithm G is a secure PRG, then the stream cipher that we have designed is semantically secure. That means it is computationally indistinguishable in the ciphertext only attack model. The proof intuition for proven disclaimer is as follows: what exactly is the difference between the one-time pad encryption scheme and encryption scheme that we are following in stream cipher. Well, the only difference in the stream cipher is the pad. The pad with which we are masking the message is pseudo random, whereas, the pad which is used in one-time pad is truly random. That is the only difference between the two encryption processes. That means, we have already proved that in the context of one-time pad, no distinguisher can distinguish apart whether the ciphertext c is an encryption of m_0 or m_1 except that probability 1/2.

We expect almost the same to hold even for the stream cipher that means no polynomial time distinguisher should be able to distinguish apart whether it is seeing an encryption of m_0 or whether it is seeing an encryption of m_1. Because if we have a distinguisher who can distinguish apart encryption of m_0 from an encryption of m_1 with a significant probability better than half that means that adversary knows how to distinguish G(k) from a uniformly random k, which is again a contradiction to the claim we are making about the security of the algorithm G.

(Refer Slide Time: 06:13)



Stream Cipher : Security Analysis

□ If $G: \{0,1\}^{\ell} \Rightarrow \{0,1\}^{L}$ is a PRG, then stream cipher is semantically-secure (computationally-indistinguishable) in the COA attack model

$G: \{0,1\}^{\ell} \Rightarrow \{0,1\}^{L}$ $\Pi_{sc} = (\text{Gen}_{sc}, \text{Enc}_{sc}, \text{Dec}_{sc})$ $\Pi_{OTP} = (\text{Gen}_{OTP}, \text{Enc}_{OTP}, \text{Dec}_{OTP})$

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

□ $b = 0 : y \in_r \{0,1\}^{L}$

□ $b = 1 : y = G(s)$, where $s \in_r \{0,1\}^{\ell}$

□ Distinguishing advantage of $\mathcal{D}$:

❖ Pr[$\mathcal{D}$ outputs $b' = 1 \mid b = 0$] = Pr[$\mathcal{A}$ outputs $b'' = B$ in COA indistinguishability game against Π_{OTP}] = ½

So, we formally capture this intuition through a reduction-based proof. So, we are given the publicly known pseudo random generator, and we consider two encryption process one, the stream cipher and the second encryption process is the one-time pad process.

Assume for the moment you have an algorithm A, who can win the indistinguishability game in the ciphertext only attack model against your stream cipher. That means we are making an assumption that our stream cipher is not semantically secure. By that we mean we have an adversary A who can win the ciphertext only attacked model indistinguishability game with probability half plus non negligible. Using the help of the algorithm A we want to design another algorithm D, who can distinguish apart a pseudo random sample produced by G from a truly random sample, which will contradict the claim that we are making about algorithm G.

So, the algorithm D participates in an instance of the indistinguishability based game for the PRG, where it is given sample y$\in\{0, 1\}^{L}$ bits and it has to find out whether y is generated uniformly randomly or whether the sample y is generated by running the algorithm G.

Now, what D does is it participates in an instance of the COA indistinguishability game against the adversary A. So, my distinguisher D is now playing a dual role here. On the left-hand side part of the experiment of the reduction it is actually participating as the distinguisher and trying to distinguish apart whether y is pseudo random or truly random, whereas in the right-hand side part of my reduction D is actually participating as a verifier of the COA indistinguishability based experiment.

So, as part of the COA based indistinguishability-based experiment adversary A throws a pair of plain text messages m_0, $m_1 \in \{0, 1\}^{L}$ as per the choice of the adversary, and the distinguisher D has to randomly encrypt one of these 2 messages, either m_0 or m_1. It decides the index of the message which I denoted by $B \leftarrow \{0, 1\}$ and with probability 1/2.

Now to encrypt the message m_B, it has to select a pad. So, what distinguisher D does it uses the sample y which is given as a challenge for the distinguisher as the pad and it masked the message m_b with the challenge y and produces the ciphertext $c = m_B \oplus y$ and gives it to the adversary A. So, now if you see what exactly is happening here is if the challenge sample y which is given to the distinguisher D is uniformly random, then what the distinguisher D has created here for the adversary is an instance of the OTP based indistinguishability experiment. Because in that case the challenge ciphertext would be an encryption of the message m_B as per an instance of one-time pad, because y would be truly random.

On the other hand, if the challenge sample y is pseudo random, in that case, D basically has created an instance of indistinguishability COA based indistinguishability experiment as per the stream cipher because in that case the challenge ciphertext c would look like a challenge ciphertext which adversary would have seen by participating in an instance of COA indistinguishability experiment against the stream cipher. Notice that the distinguisher D does not know whether it has created any instance of OTP experiment or whether it has created an instance of stream cipher experiment.

Now, depending upon what type of ciphertext c is given to the adversary, adversary A gives an output that means it tells whether the ciphertext c is an encryption of messages m_0 or whether it is an encryption of m_1. Now, using the response of adversary A, D has to find out what exactly is the type of y whether it is truly random or whether it is pseudo random. So, here is the decision output of my algorithm D.

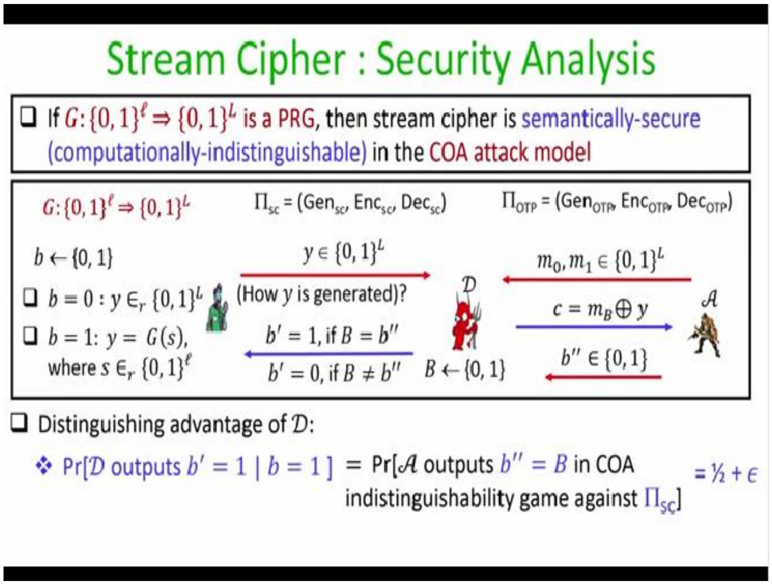
If it sees that the adversary A has correctly identified what is encrypted in c, that means b’’ = B, then the distinguisher D labels the sample y as if it is generated by a pseudo random generator. Otherwise it labels the sample y as if it is generated by a truly random generated. That is what is the idea of this distinguisher.

Now let us calculate the distinguishing advantage of the distinguisher. Let us first calculate what is the probability that the distinguisher D we have constructed labels truly random sample y as an outcome of a pseudo random sample. Well, this is exactly the same probability with which an adversary A participating in the COA indistinguishability experiment against one-time pad would have won the experiment.

Because if B = 0, that means the sample y is truly random, then the challenge ciphertext c which is given to the adversary is as per an instance of OTP. And with whatever probability the adversary A would have won the indistinguishability game against OTP. Only in that case, the distinguisher D would have output b’ = 1 because remember, the strategy of D outputting b’ = 1 is if my adversary A can win the indistinguishability game.

So, if A can win the indistinguishability game against an instance of OTP, then D would have labelled a truly random sample as a pseudo random sample. We know that the probability with which the A could win the COA indistinguishability game against an instance of OTP =1/2.

(Refer Slide Time: 12:21)



Stream Cipher : Security Analysis

□ If $G:\{0,1\}^\ell \Rightarrow \{0,1\}^L$ is a PRG, then stream cipher is semantically-secure (computationally-indistinguishable) in the COA attack model

$G:\{0,1\}^\ell \Rightarrow \{0,1\}^L$ $\Pi_{sc} = (\text{Gen}_{sc}, \text{Enc}_{sc}, \text{Dec}_{sc})$ $\Pi_{OTP} = (\text{Gen}_{OTP}, \text{Enc}_{OTP}, \text{Dec}_{OTP})$

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

□ $b = 0 : y \in_r \{0,1\}^L$

□ $b = 1$: $y = G(s)$, where $s \in_r \{0,1\}^\ell$

$y \in \{0,1\}^L$

(How y is generated)?

$b' = 1$, if $B = b''$

$b' = 0$, if $B \neq b''$

$\mathcal{D}$ $B \leftarrow \{0,1\}$

$m_0, m_1 \in \{0,1\}^L$

$c = m_B \oplus y$

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

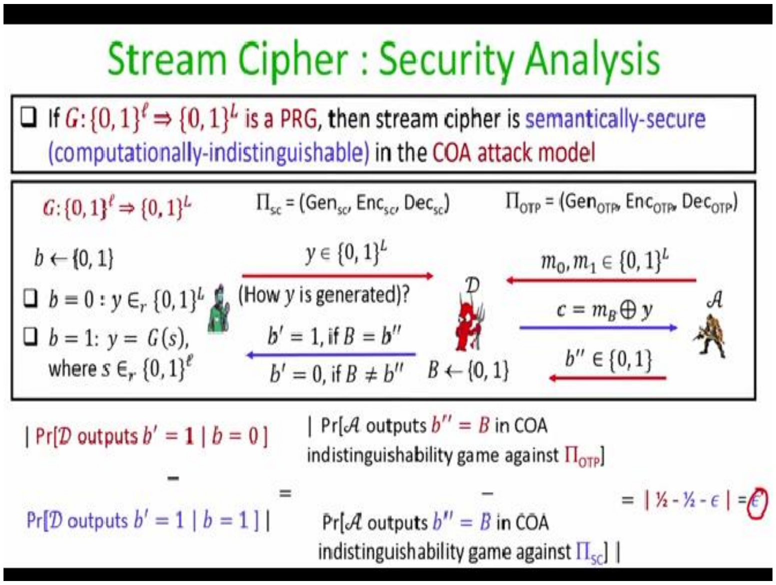
$\mathcal{A}$

□ Distinguishing advantage of $\mathcal{D}$:

❖ Pr[$\mathcal{D}$ outputs $b' = 1 \mid b = 1$] = Pr[$\mathcal{A}$ outputs $b'' = B$ in COA indistinguishability game against Π_{SC}] = ½ + ϵ

On the other hand the probability that D outputs b’ = 1 / b = 1 is exactly the same with which my adversary A can win the COA indistinguishability game against an instance of stream cipher and as per my assumption, it is 1/2 + some non-negligible probability.

(Refer Slide Time: 12:43)



Stream Cipher : Security Analysis

□ If $G:\{0,1\}^\ell \Rightarrow \{0,1\}^L$ is a PRG, then stream cipher is semantically-secure (computationally-indistinguishable) in the COA attack model

$G:\{0,1\}^\ell \Rightarrow \{0,1\}^L$ $\Pi_{sc} = (\text{Gen}_{sc}, \text{Enc}_{sc}, \text{Dec}_{sc})$ $\Pi_{OTP} = (\text{Gen}_{OTP}, \text{Enc}_{OTP}, \text{Dec}_{OTP})$

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

□ $b = 0 : y \in_r \{0,1\}^L$

□ $b = 1$: $y = G(s)$, where $s \in_r \{0,1\}^\ell$

$y \in \{0,1\}^L$

(How y is generated)?

$b' = 1$, if $B = b''$

$b' = 0$, if $B \neq b''$

$\mathcal{D}$ $B \leftarrow \{0,1\}$

$m_0, m_1 \in \{0,1\}^L$

$c = m_B \oplus y$

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

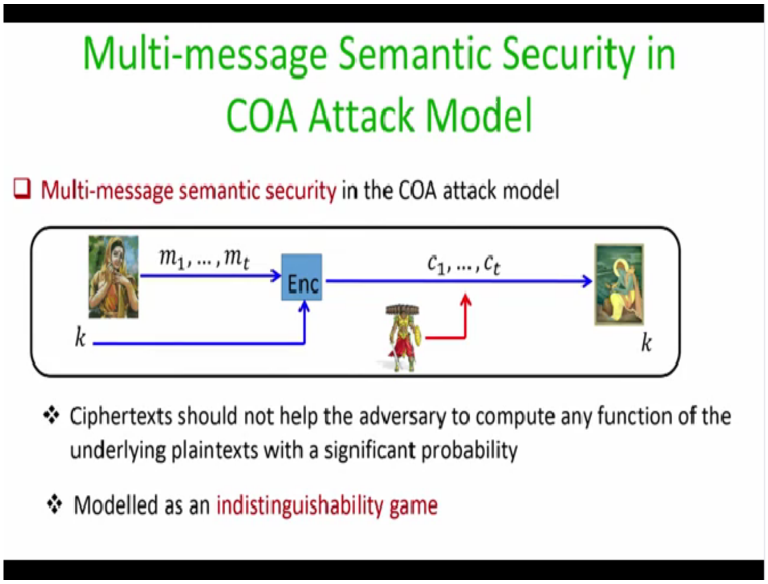
$\mathcal{A}$

| Pr[$\mathcal{D}$ outputs $b' = 1 \mid b = 0$] − Pr[$\mathcal{D}$ outputs $b' = 1 \mid b = 1$] | = | Pr[$\mathcal{A}$ outputs $b'' = B$ in COA indistinguishability game against Π_{OTP}] − Pr[$\mathcal{A}$ outputs $b'' = B$ in COA indistinguishability game against Π_{SC}] | = | ½ - ½ - ϵ | = ϵ

So, overall the distinguishing advantage of my distinguisher that we have constructed is the distinguishing advantage or the difference in the absolute probability with which the adversary A could win the COA indistinguishability game against an instance of OTP and against an instance of stream cipher, and if you take the absolute difference of these 2 probabilities it turns out to be ϵ.

So, if I assume ϵ to be non negligible probability, then it implies that I have constructed a distinguisher D who can distinguish apart a truly random sample from a pseudo random sample with the same probability and that will contradict my assumption that algorithm G is a secure PRG.

(Refer Slide Time: 13:28)



Multi-message Semantic Security in COA Attack Model

❑ Multi-message semantic security in the COA attack model

❖ Ciphertexts should not help the adversary to compute any function of the underlying plaintexts with a significant probability

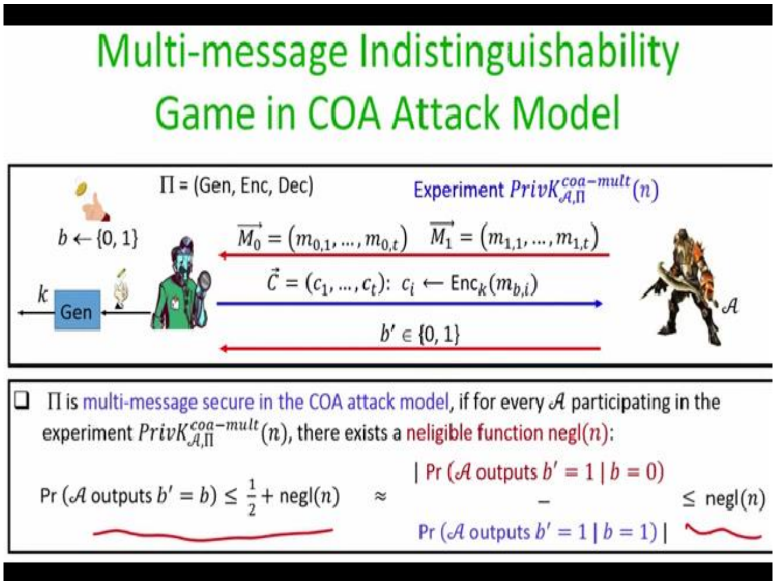
❖ Modelled as an indistinguishability game

So, even though we have now solved one of the problems or the limitations of perfect secrecy, namely, we have seen a mechanism, namely stream cipher where we could encrypt arbitrary long messages using short keys, it turns out that we cannot get rid of the second restriction. That means we cannot encrypt multiple messages using the same key with the help of stream cipher.

So, for this recall the single message semantic security. In the single message semantic security, the goal was that sender encrypted a single message using a short key. And by eavesdropping the cipher text, we wanted to ensure that adversary could not compute any function of the underlying plain text.

Whereas for the multi message semantic security, the requirement is that now sender would like to encrypt a sequence of messages using the same key and cipher text are eavesdropped by the adversary. And we want to capture the intuition that seeing the cipher text does not help the adversary to compute any function of the underlying plain text with a significant probability. And we can model this requirement as an indistinguishability game.

(Refer Slide Time: 14:35)



Multi-message Indistinguishability Game in COA Attack Model

Π = (Gen, Enc, Dec)

Experiment $PrivK^{coa-mult}_{\mathcal{A},\Pi}(n)$

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

$\overrightarrow{M_0} = (m_{0,1}, \ldots, m_{0,t})$ $\overrightarrow{M_1} = (m_{1,1}, \ldots, m_{1,t})$

$\vec{C} = (c_1, \ldots, c_t)$: $c_i \leftarrow \mathsf{Enc}_k(m_{b,i})$

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

❑ Π is multi-message secure in the COA attack model, if for every $\mathcal{A}$ participating in the experiment $PrivK^{coa-mult}_{\mathcal{A},\Pi}(n)$, there exists a neligible function negl(n):

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

So, in this indistinguishability game which we call as Experiment $PrivK^{coa-mult}_{\mathcal{A},\Pi}(n)$ because we want to capture the multi message security experiment. The rules of the games are almost identical as well as it was in the single message indistinguishability game. The difference now is instead of sending a pair of messages to the verifier; the adversary will now submit a pair of vector of messages of polynomial size.

And this basically captures the scenario that the adversary would like to distinguish apart where the sender has encrypted one vector of message or the other vector of message. So, there is no restriction on what type of messages adversary can put in these two vectors. The only restriction is that the overall size of the vectors should be a polynomial function of your security parameter.

$\overrightarrow{M_0} = (m_{0,1}, \ldots, m_{0,t})$ and $\overrightarrow{M_1} = (m_{1,1}, \ldots, m_{1,t})$ such that $|m_{0,i}| = |m_{1,i}|$. Now, what the verifier is going to do is, it will run the key generation algorithm and obtains a uniformly random key. With the help of the key, it is going to encrypt all the messages in one of the vectors. That means it will randomly choose one of the vectors with probability 1/2, it could be either the 0th vector or the first vector.

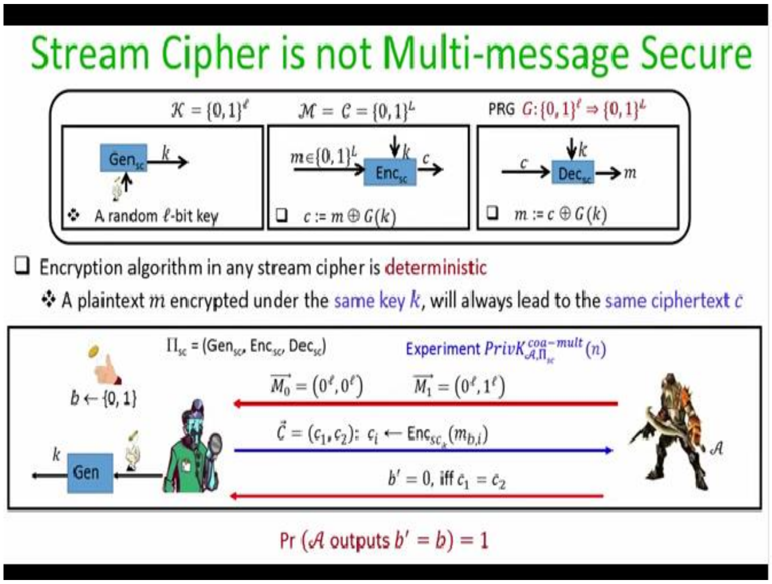
And it is going to encrypt all the messages in that vector. The challenge for the attacker is to find out by seeing the ciphertext, which vector has been encrypted whether it is the messages in the 0th vector or whether it is the messages in the first vector.

The definition of a multi message security is we will say that our encryption process is multi message secure in the COA attack model for any polynomial time adversary participating in this experiment, there exist a negligible function such that the probability with which our adversary A could output b = b'. That means it can correctly identify which vector has been encrypted is upper bounded by some half plus negligible probability function in the security parameter. Again why 1/2 because there is always a guessing adversary who can just guess which vector has been encrypted.

The extra negligible function here is to model the fact that we are in the computationally secure world. So, we are willing to give the adversary an extra additional negligible chance that it can break our scheme. An alternate security definition for the multi message security in the COA attack model is that the distinguishing advantage of any polynomial time algorithm participating in this experiment should be upper bounded by a negligible function.

That means irrespective of whether it was the 0th vector which is encrypted, or whether it is the first vector, which is encrypted in the challenge ciphertext c, the response of the adversary A should be almost the same, except with some negligible probability. So, either we can use this definition, or we can use this definition it turns out that both these conditions are equivalent to each other.

(Refer Slide Time: 17:44)



Stream Cipher is not Multi-message Secure

$\mathcal{K} = \{0,1\}^\ell$ $\mathcal{M} = \mathcal{C} = \{0,1\}^L$ PRG $G: \{0,1\}^\ell \Rightarrow \{0,1\}^L$

$\text{Gen}_{sc} \xrightarrow{k}$

❖ A random ℓ-bit key

$m \in \{0,1\}^L \rightarrow \text{Enc}_{sc} \xrightarrow{c}$

❑ $c := m \oplus G(k)$

$\xrightarrow{c} \text{Dec}_{sc} \rightarrow m$

❑ $m := c \oplus G(k)$

❑ Encryption algorithm in any stream cipher is deterministic

❖ A plaintext m encrypted under the same key k, will always lead to the same ciphertext c

$\Pi_{sc} = (\text{Gen}_{sc}, \text{Enc}_{sc}, \text{Dec}_{sc})$

Experiment $PrivK^{coa-mult}_{\mathcal{A},\Pi_{sc}}(n)$

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

$\overrightarrow{M_0} = (0^\ell, 0^\ell)$ $\overrightarrow{M_1} = (0^\ell, 1^\ell)$

$\vec{C} = (c_1, c_2)$: $c_i \leftarrow \text{Enc}_{sc_k}(m_{b,i})$

k Gen

$b' = 0$, iff $c_1 = c_2$

$\mathcal{A}$

Pr ($\mathcal{A}$ outputs $b' = b$) = 1

So, now, let us quickly see that why stream cipher is not multi message secure. So here you are given a stream cipher. And main reason that your stream cipher is not multi message security is that the encryption process and the stream cipher is that deterministic algorithm. That means if you encrypt the same message m using the same key multiple times you are going to get the same ciphertext c. That is basically loophole which any adversary could exploit in the multi message security experiment.

So, what I am going to show you is an instance of an experiment where any adversary can break the notion of multi message security against your stream cipher. What the adversary is going to do is it is going to submit a pair of vectors where $\overrightarrow{M_0} = (0^\ell, 0^\ell)$ and $\overrightarrow{M_1} = (0^\ell, 1^\ell)$. And as per the rules of the experiment, the verifier is going to randomly going to encrypt the messages in one of these 2 vectors, by using a key generated by the key generation algorithm. And once the adversary sees the challenge ciphertext, it is very simple for the adversary to pinpoint whether it is seeing the encryption of the message in the 0th vector or whether it is seeing the messages in the first vector.

Basically, strategy will be $b' = 0,$ iff $c_1 = c_2$. And this basically coming from the fact that your cipher is deterministic. So, an encryption of the message all 0s is always going to produce the same ciphertext if it is encrypted multiple times with the same key.

So, that brings me to the end of this lecture. So, in this lecture, what we have seen is we have seen that using the help of pseudo random generator. We can encrypt arbitrary long messages using short keys. And this is done through the help of stream ciphers. However, stream cipher does not give us the security against multiple messages, that means it does not allow you to reuse the same key multiple times. I hope you enjoyed this lecture, thank you!