CSA E0 235: Cryptography February 15, 2016

Scribe for Lecture 10

Instructor: Arpita Patra *Submitted by: Kshitij Bhardwaj*

1 Outline of the Scribe

- **Fallout from last lecture** - Some Rectifications.
- **Authenticated Encryption(AE)**
  - Construction of AE from cpa-secure SKE and scma-secure MAC.
  - Proof of the construction given above.
  - Does a similar reduction hold for Authenticate-then-Encrypt?
  - Need for Independent keys for Enc and MAC.
  - AE implies cca-secure SKE.
- **Conclusion.**
- **References.**

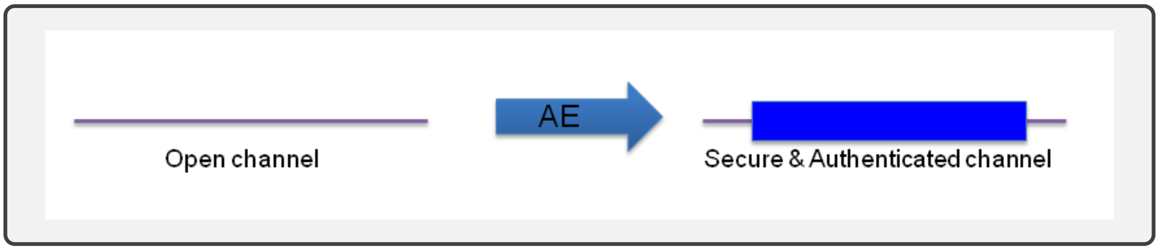
2 Fallout from last lecture

It was informed that there was a rectification to be made in the previous lecture and the same is given below: -

- **It is NOT true that you need randomized MAC to satisfy scma-security**
- Any MAC that has **canonical verification** and is cma-secure, will inherently be scma-secure.
- **Canonical Verification:** For deterministic mesaage authentication codes (that is, where tag generation is a deterministic algorithm), the canonical way to perform verification is to simply re-compute the tag and check for equality. In other words, the verify algorithm first computes tag again on the message recieved and then outputs 1 if and only if computed tag is same as the recieved tag. However, even for deterministic MACs, it is useful to define a separate verify algorithm in order to explicitly distinguish the semantics of *authenticating* a message vs. *verifying* its authenticity.
- Every deterministic MAC has canonical verification and for a deterministic MAC it is good enough to prove cma-security as scma security becomes inherently `free´. Following the same reasoning, PRF-based scheme will be scma-secure as it is deterministic and provably cma-secure.

3 Authenticated Encryption (AE)

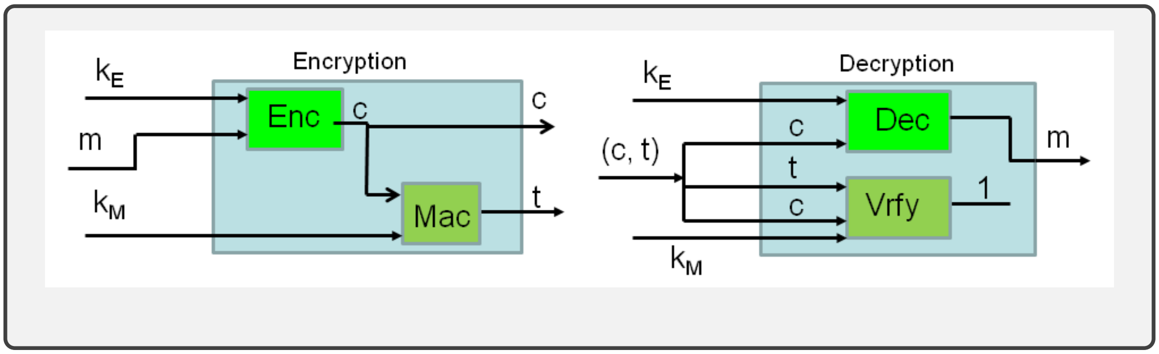
We know that it is possible to obtain *secrecy* in the private-key setting using encryption and we have also seen how to ensure *integrity* using message authentication codes. It is human nature to always ask for more and better, thus one might naturally want to achieve both goals simultaneously. In a nut shell AE can be given as under :



Our goal is to realize an "ideally secure"communication channel that provides both secrecy and integrity. To make our life simple, we define authenticated encryption by separately defining secrecy and integrity.

3.1 Construction of AE from CPA-secure SKE and SCMA-secure MAC.

To construct an authenticated encryption scheme, we must have a *CPA-secure* private-key encryption scheme (that is, no PPT attacker should be able to non-negligibly distinguish between encryption of two messages of its choice, even if it has access to encryption oracle service) and which is *SCMA-secure* or in other words has *ciphertext integrity* (that is, it is hard to come up with a ciphertext that has valid decryption even after sufficient training). We have seen both the concepts in adequate detail in the previous lectures and now we proceed to combine them to get our construction for authenticated encryption.
The way we go on to add the ingredients is given below in the form of a block diagram and is called ***"Encrypt-then-Authenticate"***.



The above given approach always leads to an AE, irrespective of the way Encryption or MAC are instantiated.

Definition 1 This AE scheme is a collection of the following three algorithms: -

- Key Generation Algorithm (**Gen()**) - This algorithm generates two secret keys,such that $k_E \in_R \{0,1\}^n$ and $k_M \in_R \{0,1\}^n$.
- Encryption Algorithm (**Enc()**) - This algorithm takes three inputs viz. the message m & the two keys k_E and k_M and produces a tuple of the ciphertext and the tag as output in the following way: - $c \leftarrow \mathsf{Enc}_{k_E}(m)$ and $t \leftarrow \mathsf{Mac}_{k_M}(c)$.
- Decryption Algorithm (**Dec()**) - This algorithm takes as input the tuple of the ciphertext and the tag and first verifies whether the ciphertext is indeed the same ciphertext which was sent, that is, it corresponds to the tag which has been sent along, and only once it is verified will the ciphertext be decrypted and the original message is retrieved. This is done in the following way: -

```
if $Vrfy_{k_M}(c) = 0$ then
    output $\perp$
else
    $m := Dec_{k_E}(c)$
end
```

♢

3.2 Proof of Security of Scheme given in para 3.1 above.

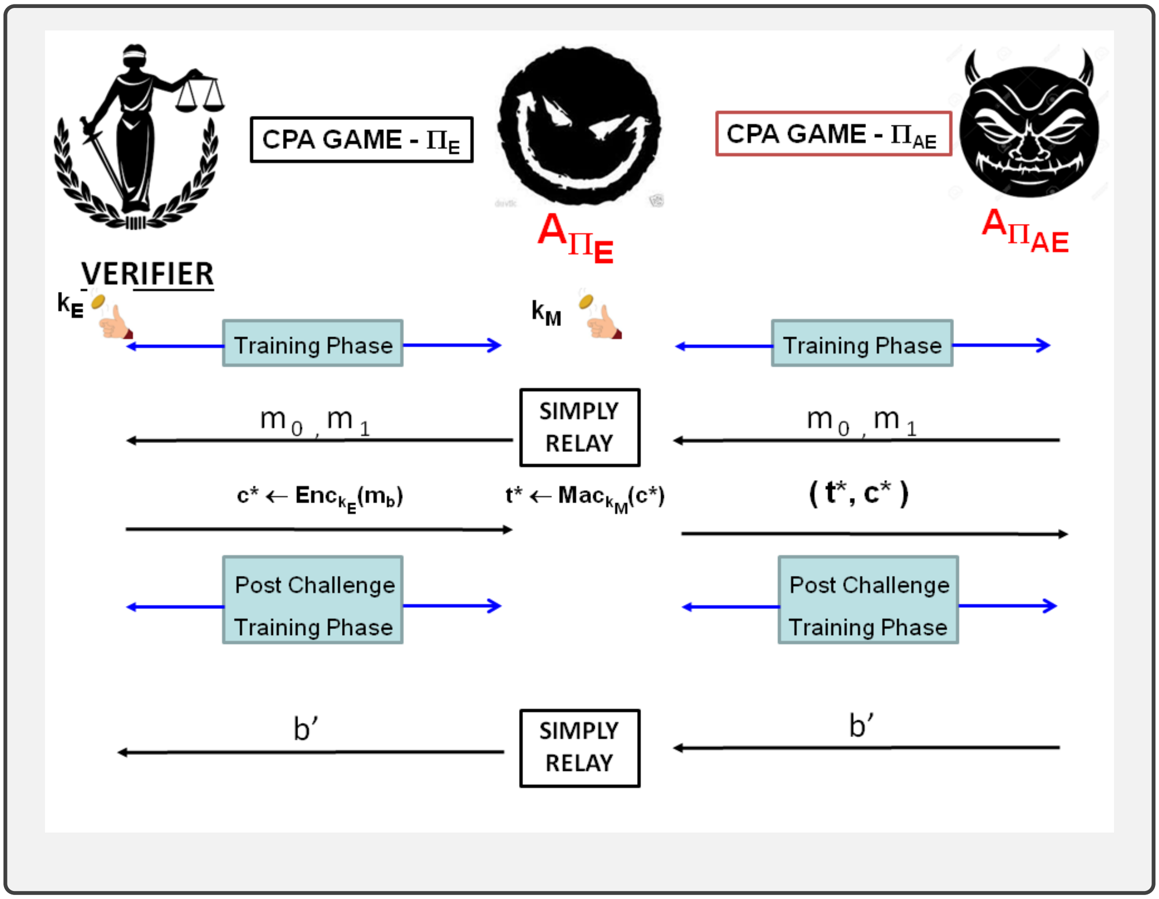
The scheme given above states that to construct a AE scheme we need the SKE to be CPA-secure and the MAC to be SCMA-secure. Let the SKE be denoted by $\pi_E = (Gen_E, Enc_E, Dec_E)$, MAC by $\pi_M = (MAC, Vrfy)$ and the resultant AE is denoted by $\pi_{AE} = (Gen_{AE}, Enc_{AE}, Dec_{AE})$ respectively.

The proof given here is on the same lines as was explained in the class with the help of the game based strategy. So, let us divide the proof into two parts as given below: -

- If π_E is CPA-secure than π_{AE} is also CPA-secure
- If π_M is SCMA-secure than π_{AE} is also CSMA-secure

3.2.1 Proof of: If π_E is CPA-secure then π_{AE} is also CPA-secure .

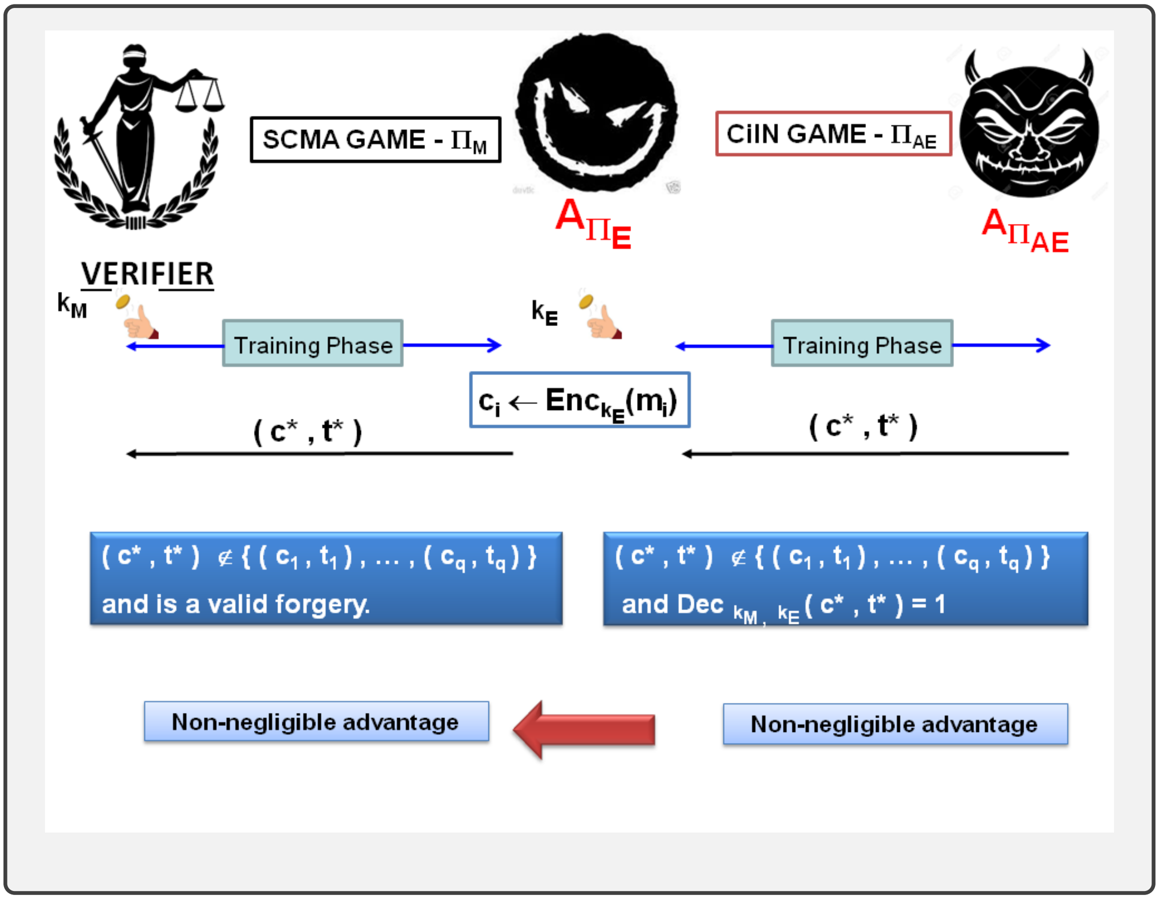
We will prove this by contradiction, so we assume that π_E is CPA-secure but π_{AE} is NOT CPA-secure. Hence, in the game based strategy we can say that there exists a polytime adversary,$A_{\pi_{AE}}$, which can win the game with a non-negligible probability and break the scheme, π_{AE}. The reduction based proof is given below in the form of a diagram: -



We see here that the A_{π_E} uses the power of $A_{\pi_{AE}}$ to break the CPA security of scheme π_E and to do this it simulates the scheme π_{AE} and plays the CPA game with the verifier. We clearly see in the game that if $A_{\pi_{AE}}$ has a non-negligible advantage then so does A_{π_E}, i.e. scheme π_E is not CPA-secure which is a contradiction to our assumption and hence we conclude by saying that scheme π_{AE} is also CPA-secure.

3.2.2 Proof of: If π_M is SCMA-secure then π_{AE} is also SCMA-secure .

We will prove this too by contradiction, so we assume that π_M is SCMA-secure but π_{AE} is NOT SCMA-secure. Hence, in the game based strategy we can say that there exists a polytime adversary,$A_{\pi_{AE}}$, which can win the game with a non-negligible probability and break the scheme, π_{AE}. The reduction based proof is given below in the form of a diagram: -


VERIFIER
SCMA GAME - Π_M
A_{Π_E}
CiIN GAME - Π_{AE}
$A_{\Pi_{AE}}$
k_M
k_E
Training Phase
Training Phase
$c_i \leftarrow Enc_{k_E}(m_i)$
(c^*, t^*)
(c^*, t^*)
$(c^*, t^*) \notin \{(c_1, t_1), \ldots, (c_q, t_q)\}$ and is a valid forgery.
$(c^*, t^*) \notin \{(c_1, t_1), \ldots, (c_q, t_q)\}$ and $Dec_{k_M, k_E}(c^*, t^*) = 1$
Non-negligible advantage
Non-negligible advantage

We see here that the A_{π_M} uses the power of $A_{\pi_{AE}}$ to break the SCMA security of scheme π_M and to do this it simulates the CiIn game for $A_{\pi_{AE}}$ and plays the SCMA game with the verifier. We clearly see in the game that if $A_{\pi_{AE}}$ has a non-negligible advantage then so does A_{π_M}, i.e. scheme π_M is not SCMA-secure which is a contradiction to our assumption and hence we conclude by saying that scheme π_{AE} is also SCMA-secure.

3.3 Does a similar reduction hold for Authenticate-then-Encrypt?

This reduction based proof can not be applied to the ***"authenticate-then-encrypt"*** scheme as the second half of the proof will not materialise in this modified case as the message and tag pair returned by the $A_{\pi_{AE'}}$ may not be a valid forgery attempt in the SCMA game being played by A_{π_M}.

3.4 Need for Independent keys for Enc and MAC.

A basic principle of cryptography: *different instances of cryptographic primitives should always use independent keys.* Cryptography is a double edged sword and more than anywhere else, half knowledge can be severely harmful in this field. A seemingly negligible change may render a perfect scheme useless.Consider the just discussed ***“encrypt-then-authenticate”*** methodology and let the two keys used for encryption and authentication be same. Let F be SPRP then it follows that F^{-1} is a SPRP too.

Definition 2 Let us define this AE scheme as follows: -

- $Enc_k(m) = F_k(m\|r)$for $m \in \{0,1\}^{n/2}$ and a uniform $r \in \{0,1\}^{n/2}$, and define $Mac_k(c) = {F_k}^{-1}(c)$.

♢

It can be shown that this Encryption scheme is CCA-secure and we know that the given authentication code is a secure MAC. However, the *“encrypt-then-authenticate”* combination using the ***same key*** k as applied to the message m yields: -

$$Enc_k(m), Mac_k(Enc_k(m)) = F_k(m\|r), {F_k}^{-1}(F_{k(m\|r)}) = F_k(m\|r), m\|r$$

As we can see this reveals the message m in clear! The only thing that was different and went wrong from the proved AE was that the keys were same instead of being chosen (uniformly and) independently.

3.5 AE implies CCA-Secure SKE.

Theorem 1 *If π_{AE} is an AE scheme, then π_{AE} is CCA-secure.*
Proof *Assume π_{AE} is not CCA-secure,so we can say that there exists an adversary $A_{\pi_{AE}}$ and a polynomial $p(n)$ such that: -*

$$Pr[Privk^{CCA}_{A_{\pi_{AE}},\pi_{AE}}(n) = 1] > 1/2 + 1/p(n)$$

As was explained in the class there are two ways to win this game: -

- *By making a Valid Query to the Decryption Oracle.*
- *By not making a Valid Query to the Decryption Oracle.*

*A ciphertext `c ´is **new** if the adversary did not receive it from its Encryption Oracle or as the challenge ciphertext. Also, let **Valid Query** be the event that the adversary submits a new ciphertext to its Decryption Oracle which is valid, i.e. for which $Vrfy_{k_M}(c,t) = 1$ and $\overline{\textbf{Valid Query}}$ be the event that the adversary doesn't submit a new ciphertext which is valid.*

Hence, we can write the above equations as: -

$$Pr[Privk^{CCA}_{A_{\pi_{AE}},\pi_{AE}}(n) = 1 \wedge ValidQuery] + Pr[Privk^{CCA}_{A_{\pi_{AE}},\pi_{AE}}(n) = 1 \wedge \overline{ValidQuery}]$$

$$\leq Pr[ValidQuery] + Pr[Privk^{CCA}_{A_{\pi_{AE}},\pi_{AE}}(n) = 1 \wedge \overline{ValidQuery}]$$

combining the above equations, we get: -

$$Pr[ValidQuery] + Pr[Privk^{CCA}_{A_{\pi_{AE}},\pi_{AE}}(n) = 1 \wedge \overline{ValidQuery}] > 1/2 + 1/p(n)$$

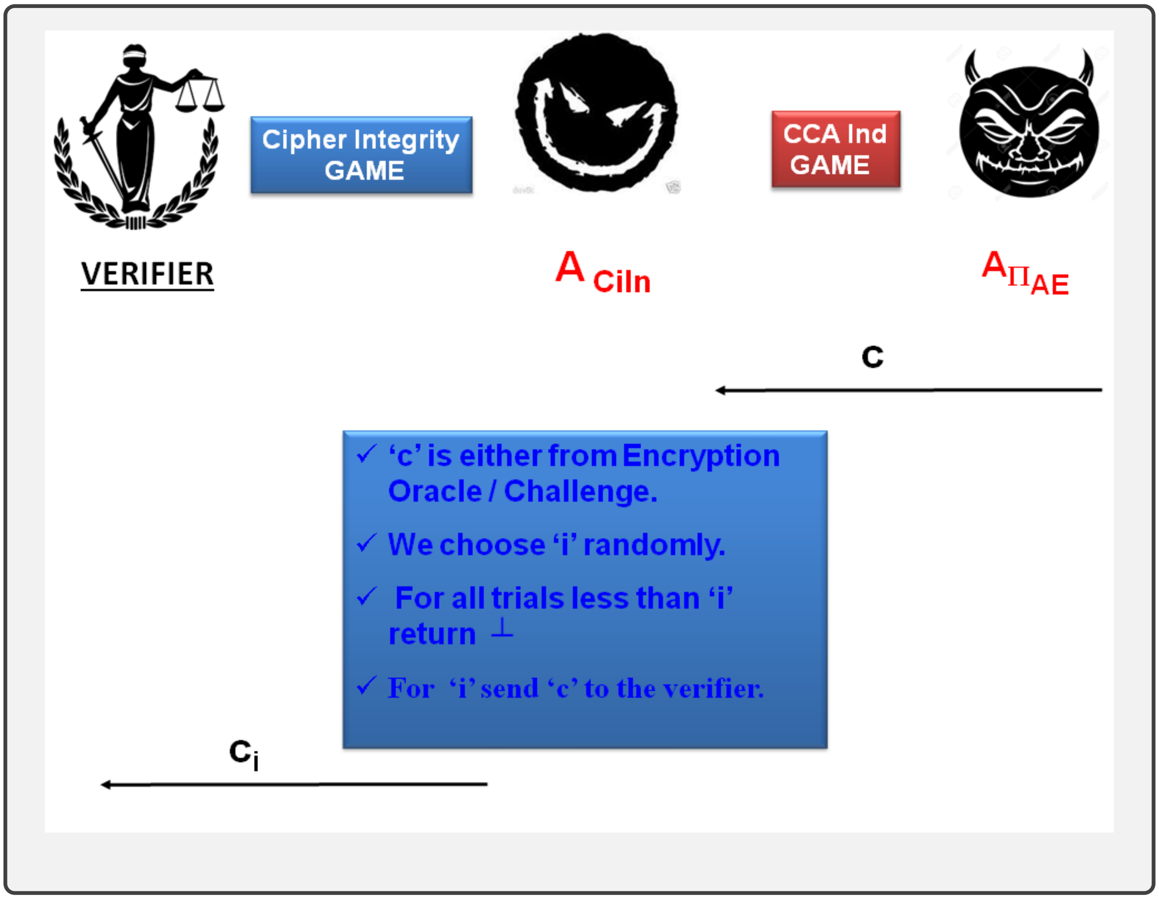
This brings us to choosing from two possible cases given below: -

- *Case 'A': $Pr[Privk^{CCA}_{A_{\pi_{AE}},\pi_{AE}}(n) = 1 \wedge \overline{ValidQuery}] > 1/2 + 1/b(x)$*
- *Case 'B': $Pr[ValidQuery] > 1/a(x)$*

From the above two cases either Case 'A'is true or Case 'B'is true.

- ***Considering Case 'A': -***
  This case basically brings out that the adversary can win with a non-negligible advantage without making a Valid Query and that would mean that it can win over the CPA security and thus the scheme can not be an Authenticated Encryption Scheme.

- ***Considering Case 'B': -***
  In the game based strategy we consider an adversary A_{CiIn} and the $A_{\pi_{AE}}$. We encash on the availability of Decryption Oracle to $A_{\pi_{AE}}$ to win the Ciphertext Integrity game.
  The encryption queries sent by$A_{\pi_{AE}}$are forwarded to the Verifier.
  To actually encash the access to Decryption Oracle A_{CiIn} chooses 'i at random and till the number of queries is not equal to 'i, either m is sent if the c was in already queried as part of training phase, or $\perp$ otherwise.
  At the i^{th} query A_{CiIn} forwards the cipher text to the verifier.
  The crux of the process after the training and challenge phase is over is shown in the diagram below: -



- ✓ 'c' is either from Encryption Oracle / Challenge.
- ✓ We choose 'i' randomly.
- ✓ For all trials less than 'i' return $\perp$
- ✓ For 'i' send 'c' to the verifier.

If i^{th} ciphertext is the first Valid Query sent by $A_{\pi_{AE}}$, then the simulation done by A_{CiIn} is perfect and the probability of A_{CiIn} winning the game is same as that of $A_{\pi_{AE}}$ sending the first Valid Query as the i^{th} query, which is as given below: -

$$Pr[CiIn_{A_{CiIn}}(n) = 1] = Pr[ValidQuery \wedge I]$$

where I is the event of forwarding the i^{th} ciphertext. Now, as I and $ValidQuery$ are independent events, therefore, we have : -

$$Pr[CiIn_{A_{CiIn}}(n) = 1] = Pr[ValidQuery]Pr[I]$$

$$\boxed{\Rightarrow Pr[CiIn_{A_{CiIn}}(n) = 1] = \frac{1}{a(x)}.\frac{1}{q(n)}}$$

*Where, $q(n)$ is the polynomial upper bound on the messages queried on Encryption Oracle.This is clearly a **non-negligible probability**.So, combining the results of both the cases we can conclude that π_{AE} is not an AE scheme, which is a contradiction and hence, our assumption is incorrect.Thus, proving that: -*

Authenticated Encryption Schemes implies CCA-Secure!

■

3.6 Conclusion.

As a conclusion for this lecture, we have the following take aways with us: -

- Every AE scheme is also a CCA-secure cipher.
- There are encryption schemes which are only CCA-secure.
- Conceptually, the goal of CCA-security and authenticated encryption are different: -
  - Aim of CCA-security is to achieve only privacy even in the case of an attacker disrupting the communication.
  - Aim of AE is to achieve both privacy as well as integrity of the message.
- With regards to the efficiency of the overall process, in the SKE world both, i.e. CCA-secure encryption scheme and AE scheme are almost equivalent, and hence, there is no reason for us to settle only for privacy and not ask for integrity as well. However, the case is not same in the Public key world, and the difference is more pronounced.

References

[1] Arpita Patra. *http://drona.csa.iisc.ernet.in/∼arpita/Cryptography16.html.*

[2] Katz, Jonathan, and Yehuda Lindell. *Introduction to Modern Cryptography.* CRC Press, 2014.