CSA E0 235: Cryptography (15 April, 2015)

Lecture 10

Instructor: Dr Arpita Patra *Submitted by : Divya Ravi*

1 Introduction

In this lecture we introduce the idea of a hybrid public-key encryption scheme which combines a public-key encryption scheme and the private-key encryption scheme. We show how this scheme can be implemented and highlight its advantage. Next we look at Key encapsulation mechanism(KEM) and prove its CPA security using a hybrid argument based proof. We also see an El Gamal like KEM based on Hash Diffie-Hellman assumption. Finally we look at the notion of CCA security in public-key setting.

2 Hybrid Encryption

We have seen Authenticated Encryption - a hybrid of private-key encryption scheme and MAC in the earlier lectures. Hybrid schemes are developed to combine best properties . Consider the two worlds of SKE and PKE . Desirable property of PKE is that no assumption of secret key is required for secure comunication as in the case of SKE. However PKE is expensive whereas SKE which uses only lightweight computation and lower ciphertext expansion is significantly faster. The resulting combination which combines the best of both is called *hybrid encryption* and is used extensively in practice.

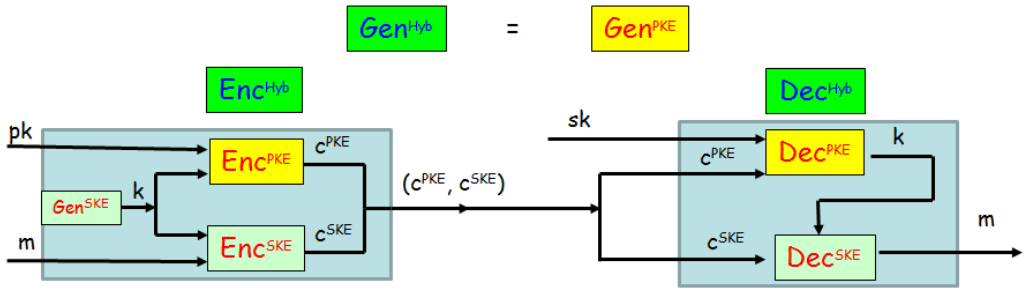

Figure 1: Direct implementation of hybrid encryption

Hybrid encryption can be implemented as shown above where the sender would share k by first choosing a uniform value k and then encypting k using public-key encryption scheme. The *Key-Encapsulation Mecahanism* (KEM) does both these actions in a single shot as we shall see later. In the above implementation suppose $(\mathsf{Gen}^{\mathsf{PKE}}, \mathsf{Enc}^{\mathsf{PKE}}, \mathsf{Dec}^{\mathsf{PKE}})$ $(\mathsf{Gen}^{\mathsf{SKE}}, \mathsf{Enc}^{\mathsf{SKE}}, \mathsf{Dec}^{\mathsf{SKE}}), (\mathsf{Gen}^{\mathsf{Hyb}}, \mathsf{Enc}^{\mathsf{Hyb}}, \mathsf{Dec}^{\mathsf{Hyb}})$ denote the PKE , SKE and Hybrid encryption schemes respectively. Then,

Implementation of Hybrid Encryption

- $\mathsf{Gen}^{\mathsf{Hyb}}$ is same as $\mathsf{Gen}^{\mathsf{PKE}}$.
- In $\mathsf{Enc}^{\mathsf{Hyb}}$, first $\mathsf{Gen}^{\mathsf{SKE}}$ generates a uniform key k which is encrypted using the public key pk by $\mathsf{Enc}^{\mathsf{PKE}}$ to generate first part of the ciphertext c^{PKE}. Next $\mathsf{Enc}^{\mathsf{SKE}}$ encrypts the message m using the private key k and generates the second part of the ciphertext c^{PKE} as shown.
- Decryption ,$\mathsf{Dec}^{\mathsf{Hyb}}$ uses the secret key, first k is obtained from c^{PKE} using $\mathsf{Dec}^{\mathsf{PKE}}$ and then this k is used by $\mathsf{Dec}^{\mathsf{SKE}}$ to decrypt the message m from c^{SKE}.

Let us now analyze the efficiency of the above hybrid encyption scheme . Let α denote the cost of encrypting a single-bit message using PKE and let β denote the cost of encrypting a single bit message using SKE. As we know, encryption using SKE is significantly faster than PKE say, α is of the order 10^5 * β. Consider the length of the message m >>>> n (i.e length of the key) . If we had used PKE scheme , then the cost per bit of plaintext would be α. If Hybrid scheme is used , the cost would include cost of encrypting n bit key using PKE and cost of encrypting m bit message using SKE. Thus the cost per bit of plaintext encrypted using Hybrid PKE is

$$\frac{n\alpha + m\beta}{m} = \frac{n\alpha}{m} + \beta \tag{1}$$

which approaches β for sufficiently long m. In the limit of very long messages, then, the cost per bit incurred by the public-key hybrid encryption scheme is the same as the cost per bit of the PKE. Hybrid encryption thus allows us to achieve the functionality of PKE at the efficiency of SKE , atleast for sufficiently long messages.

A similar calculation can be used to measure the effect of hybrid encryption on cipher text length. For some fixed value of n , let L denote the length of the ciphertext output by $\mathsf{Enc}^{\mathsf{PKE}}$ and say $\mathsf{Enc}^{\mathsf{SKE}}$ results in a ciphertext of length m + n (where m is the size of the message) . The total length of the ciphertext output by $\mathsf{Enc}^{\mathsf{Hyb}}$ is

$$L + n + m \tag{2}$$

We now compare to the case when PKE was used to encrypt the message of size m. Suppose we had used block-by-block encryption , then $\mathsf{Enc}^{\mathsf{PKE}}$ would result in ciphertext of length L * $\frac{m}{n}$ which will be greater than L + n + m for sufficiently long m.

2.1 KEM and DEM

A KEM has three algorithms similar in spirit to those of a public-key encryption scheme. As before , the key generation algorithm Gen is used to generate a pair of public and private keys . In place of encryption , we now have an encapsulation algorithm Encaps that

takes only public key as input (and no message) , and outputs a ciphertext along with a key k. A corresponding decapsulation algorithm Decaps is run by the receiver to recover k from the ciphertext c using the private key. We present a formal definition of KEM below :

Definition 1 A key-encapsulation mechanism (KEM) is a tuple of probabilistic polynomial-time algorithms (Gen, Encaps, Decaps) such that :

- The key-generation algorithm Gen takes as input the security parameter 1^n and outputs a public-/private key pair (pk,sk) . We assume pk and sk each has length at least n , and that n can be determined from pk.
- The encapsulation algorithm Encaps takes as input a public key pk and the security parameter 1^n . It outputs a ciphertext c and a key k $\in \{0,1\}^{l(n)}$ where l is the key length. We write this as (c,k) $\leftarrow$ Encaps(1^n).
- The deterministic decapsulation algorithm Decaps takes as input a private key sk and a ciphertext c , and outputs a key k or a special symbol denoting failure. We write this as k := Decaps$_{sk}$(c).

  It is required that with all but negligible probability over (sk,pk) , output by Gen(1^n) , if Encaps$_{pk}$ (1^n) outputs (c,k) then Decaps$_{sk}$(c) outputs k.

$\diamondsuit$

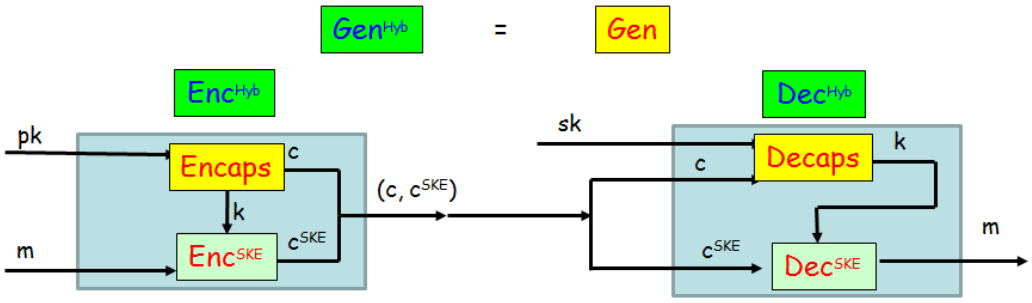

Figure 2: KEM/DEM approach

The above figure shows how KEM is implemented. The sender runs Encaps (1^n) to obtain c along with a key k , it then uses a private-key encryption scheme to encrypt its message m,using k as the key. In this context, the private-key encryption scheme is called a *data-encapsulation mechanism*(DEM). The ciphertext sent to the receiver includes both c and the ciphertext c^{SKE} from the private key scheme. The formal construction is as given below.

Implementation using KEM/DEM

Let π = (Gen, Encaps, Decaps) be a KEM with key length n, and let π^{SKE} be a private-key encryption scheme. A hybrid public-key encryption scheme $\pi^{\mathsf{Hyb}} = (\mathsf{Gen}^{\mathsf{Hyb}}, \mathsf{Encaps}^{\mathsf{Hyb}}, \mathsf{Decaps}^{\mathsf{Hyb}})$ is constructed as follows :

- $\mathsf{Gen}^{\mathsf{Hyb}}$: on input 1^n run $\mathsf{Gen}(1^n)$ and use the public and private keys (pk,sk) that are output.
- $\mathsf{Enc}^{\mathsf{Hyb}}$: on input a public key pk and a message m $\in \{0,1\}^*$ do :
  1. Compute (c, k) $\leftarrow \mathsf{Encaps}(1^n)$
  2. Compute $\mathrm{c}^{\mathsf{SKE}} \leftarrow \mathsf{Enc}_k^{\mathsf{SKE}}(m)$
  3. Output the ciphertext $\langle c, c^{\mathsf{SKE}} \rangle$
- $\mathsf{Dec}^{\mathsf{Hyb}}$: On input a private key sk and a ciphertext $\langle c, c^{\mathsf{SKE}} \rangle$ do :
  1. Compute k := $\mathsf{Decaps}(c)$
  2. Output the message m := $\mathsf{Dec}_k^{\mathsf{SKE}}(c^{\mathsf{SKE}})$

2.2 CPA security for KEM

In the notion of CPA security as seen earlier for PKE / SKE , the challenge for the adversary was to distinguish whether a ciphertext c is an encryption of some message m_0 or some other message m_1. In case of KEM , there is no message - Here the challenge for the adversary is to distinguish between an encapsulated key k and a uniform key that is independent of the ciphertext c.Let $\pi = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ be a KEM and A an arbitrary adversary.The CPA indistinguishability experiment is formalized as

CPA indistinguishability experiment $\mathsf{KEM}^{\mathsf{cpa}}_{\mathsf{A},\pi}(n)$ **:**

- $\mathsf{Gen}(1^n)$ is run to obtain keys (pk,sk). Then $\mathsf{Encaps}_{pk}(1^n)$ is run to generate (c,k) with k $\in \{0,1\}^n$
- A uniform bit b $\in \{0,1\}$ is chosen. If b = 0 , set k' := k . If b = 1 then choose a uniform k' $\in \{0,1\}^n$.
- Give (pk, c, k') to A, who outputs a bit b'. The output of the experiment is defined to be 1 if b' = b , and 0 otherwise.

In the experiment , A is given the ciphertext c and either the actual key corresponding to c , or an independent uniform key. The KEM is CPA-secure if no efficient adversary can distinguish between these possibilities. CPA security of KEM is formalized below.

Definition 2 A key-encapsulation mechanism π is CPA-secure if for all probabilistic polynomial-time adversaries A there exists a negligible function negl such that

$$Pr[\mathsf{KEM}^{cpa}_{\mathsf{A},\Pi}(n) = 1] \leq \frac{1}{2} + negl(n) \tag{3}$$

$\diamondsuit$

CPA security of public key encryption scheme can be achieved by a hybrid of a CPA secure KEM and a SKE that has indistinguishable encryptions in the presence of an eavesdropper (COA secure). We shall see this in further detail in the following theorem

Theorem 1 *(Blum Goldwasser CRYPTO'84):Hybrid π^{Hyb} of a CPA-secure KEM π and a COA-secure SKE π^{SKE} is a CPA-secure PKE.*

Proof The proof uses a standard indistinguishability based hybrid argument. Let the notation X $\equiv$ Y denote the event that no polynomial time adversary can distinguish between two distributions X and Y. π is CPA-secure KEM means that

$$(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Encaps}^{(2)}_{pk}(1^n)) \equiv (pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), k')$$

Here $\mathsf{Encaps}^{(1)}_{pk}(1^n)$, $\mathsf{Encaps}^{(2)}_{pk}(1^n)$ denotes the cipherext and key output by Encaps respectively. pk is generated by $\mathsf{Gen}(1^n)$ and k' is chosen independently and uniformly from $\{0,1\}^n$.Similarly the fact that π^{SKE} is COA secure means that for any m_0 , m_1 output by A we have $\mathsf{Enc}^{\mathsf{SKE}}_k(m_0) \equiv \mathsf{Enc}^{\mathsf{SKE}}_k(m_1)$ if k is chosen uniformly at random. In order to prove CPA-security of π^{hy} we need to show that

$$(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_k(m_0)) \equiv (pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_k(m_1)) \tag{4}$$

The proof proceeds in three steps.

1. First we prove that

   $$(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_k(m_0)) \equiv (pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0)) \tag{5}$$

   where on the left k is output by $\mathsf{Encaps}^{(2)}_{pk}(1^n)$ and on the right k' is an independent uniform key. This follows from a straightforward reduction , since CPA security of π means that $\mathsf{Encaps}^{(2)}_{pk}(1^n)$ cannot be distinguished from a uniform key k' even given pk and $\mathsf{Encaps}^{(1)}_{pk}(1^n)$.

2. Next we prove that

   $$(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0)) \equiv (pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) \tag{6}$$

   Here the difference is between encrypting m_0 or m_1 using π^{SKE} and a uniform independent key k'. This result follows from the COA security of π^{SKE}.

3. Finally exactly similar to the first case , we prove that

   $$(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_k(m_1)) \equiv (pk, \mathsf{Encaps}^{(1)}_{pk}(1^n), \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) \tag{7}$$

   This follows again from CPA security of π

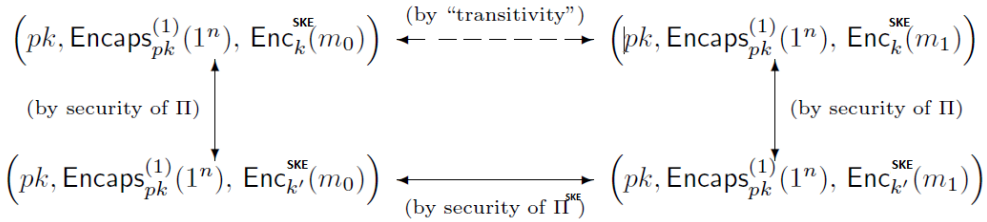

Figure 3: High level Structure of Hybrid argument based proof of theorem 1

Equations (5) ,(6) and (7) imply by transitivity the desired result that is equation (4). The figure 3 gives a high level picture of the proof. We now formalize this intutive argument .

To prove the CPA security of the hybrid public-key encryption scheme , our goal is to prove that there is a negligible function negl such that

$$Pr[\mathsf{PubK}^{\mathsf{eav}}_{\mathsf{A}^{\mathsf{hy}},\Pi^{\mathsf{hy}}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n) \tag{8}$$

where $\mathsf{PubK}^{\mathsf{eav}}_{\mathsf{A}^{\mathsf{hy}},\Pi^{\mathsf{hy}}}(n)$ denotes the experiment and A^{hy} is an arbitrary PPT adversary. By definition of the experiment , we have

$$\begin{aligned} Pr[\mathsf{PubK}^{\mathsf{eav}}_{A^{\mathsf{hy}},\Pi^{\mathsf{hy}}}(n) = 1] = \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n), \mathsf{Enc}_{k'}^{\mathsf{SKE}}(m_0)) = 0] \\ + \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n), \mathsf{Enc}_{k'}^{\mathsf{SKE}}(m_1)) = 1] \end{aligned} \tag{9}$$

where in each case k equals $\mathsf{Encaps}_{pk}^{(2)}(1^n)$. Consider the following PPT adversary A_1 attacking Π .

Adversary A_1 :

- A_1 is given (pk, c, k")
- A_1 runs $\mathsf{A}^{\mathsf{hy}}(pk)$ to obtain two messages m_0 and m_1 .Then A_1 computes c' $\leftarrow \mathsf{Enc}_{k''}^{\mathsf{SKE}}(\mathrm{m}_0)$) and gives ciphertext $\langle$ c, c' $\rangle$ to A^{hy} and then outputs the bit b' that A^{hy} outputs.

Consider the behavior of A_1 when attacking π in the experiment $\mathsf{KEM}^{\mathsf{cpa}}_{\mathsf{A}_1,\Pi}(n)$. When $\mathrm{b} = 0$ in the experiment , then A_1 is given $\langle$pk, c, k"$\rangle$ where c and k" were output by $\mathsf{Encaps}_{pk}^{(1)}(1^n)$. This means that A^{hy} is given a ciphertext of the form $\langle$c, c' $\rangle$ = $\langle$c, $\mathsf{Enc}_k^{\mathsf{SKE}}(\mathrm{m}_0)$ $\rangle$ where k is encapsulated by c. So

$$Pr[A_1 \text{outputs } 0 | b = 0] = Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n), \mathsf{Enc}_k^{\mathsf{SKE}}(m_0)) = 0]$$

On the other hand , when $\mathrm{b} = 1$ in the experiment $\mathsf{KEM}^{cpa}_{\mathsf{A}_1,\Pi}(n)$ then A_1 is given $\langle$ pk, c, k"$\rangle$ with k" uniform and independent of c. If we denote such a key by k', this means A^{hy} is

given a ciphertext of the form $\langle$ c , $\mathsf{Enc}_{k'}^{\mathsf{SKE}}(\mathrm{m}_0)$ $\rangle$ and

$$Pr[A_1 \text{ outputs } 1|b=1] = Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n, \mathsf{Enc}_{k'}^{\mathsf{SKE}}(m_0)_{=}1]$$

Since π is a CPA secure KEM, there is a negligible function negl_1 such that

$$\begin{aligned}\frac{1}{2} + \mathsf{negl}_1(n) &\geq Pr[\mathsf{KEM}_{A_1,\Pi}^{cpa}(n) = 1] \\ &= \frac{1}{2} * Pr[A_1 \text{ outputs } 0 \ |b=0] + \frac{1}{2} * Pr[A_1 \text{ outputs } 1|b=1] \\ &= \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n, Enc_{k'}^{\mathsf{SKE}}(m_0)) = 0] \\ &+ \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n, Enc_{k'}^{\mathsf{SKE}}(m_0)) = 1]\end{aligned} \tag{10}$$

where k is equal to $\mathsf{Encaps}_{pk}^{(2)}(1^n)$ and k' is a uniform and independent key .

Next, consider the following PPT adversary A' that eavesdrops on a message encrypted using the private-key scheme π^{SKE}.

Adversary A' :

- A'(1^n) runs $\mathsf{Gen}(1^n)$ on its own to generate keys (pk,sk). It also computes c $\leftarrow$ $\mathsf{Encaps}_{pk}^{(1)}(1^n$
- A' runs A^{hy}(pk) to obtain two messages m_0 and m_1 . These are output by A' , and is given a ciphertext c' in return.
- A' gives a ciphertext $\langle$ c,c'$\rangle$ to A^{hy}, and outputs the bit b' that A^{hy} outputs.

When b = 0 in experiment $\mathsf{PrivK}_{\mathsf{A}',\Pi'}^{\mathsf{eav}}(n)$, the adversary A' is given a ciphertext c' which is an encryption of m_0 using a key k' that is uniform and is independent of anything else. So A^{hy} is given a ciphertext of the form $\langle$ c,$\mathsf{Enc}_{k'}^{\mathsf{SKE}}(\mathrm{m}_0)$ $\rangle$ where k' is uniform and independent of c , and

$$Pr[\mathsf{A}' \text{ outputs } 0|b=0] = Pr[\mathsf{A}^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n, \mathsf{Enc}_{k'}^{\mathsf{SKE}}(m_0)) = 0]$$

On the other hand when b = 1 in experiment $\mathsf{PrivK}_{\mathsf{A}',\Pi'}^{\mathsf{eav}}(n)$, the adversary A' is given a ciphertext c' which is an encryption of m_1 using a key k' that is uniform and is independent of anything else. So A^{hy} is given a ciphertext of the form $\langle$ c,$\mathsf{Enc}_{k'}^{\mathsf{SKE}}(\mathrm{m}_1)$ $\rangle$ where k' is uniform and independent of c , and so

$$Pr[A' \text{ outputs } 1 \ |b=1] = Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}_{pk}^{(1)}(1^n, \mathsf{Enc}_{k'}^{\mathsf{SKE}}(m_1)) = 1]$$

Since π^{SKE} has indistiguishable encryptions in the presence of an eavesdropper , there is a

negligible function negl' such that

$$\begin{aligned}\frac{1}{2} + \mathsf{negl}'(n) &\geq Pr[\mathsf{PrivK}^{\mathsf{eav}}_{A',\Pi'}(n) = 1] \\ &= \frac{1}{2} * Pr[A' \text{ outputs } 0 | b = 0] + \frac{1}{2} * Pr[A' \text{ outputs } 1 \ \| b = 1] \\ &= \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n, Enc^{\mathsf{SKE}}_{k'}(m_0)) = 0] \\ &+ \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) = 1]\end{aligned} \tag{11}$$

Proceeding exactly as we did to prove equation (10) , we can show that there is a negligible function negl_2 such that

$$\begin{aligned}\frac{1}{2} + \mathsf{negl}_2(n) &\geq Pr[\mathsf{KEM}^{cpa}_{A_2,\Pi}(n) = 1] \\ &= \frac{1}{2} * Pr[A_2 \text{ outputs } 0 \ | b = 0] + \frac{1}{2} * Pr[A_2 \text{ outputs } 1 | b = 1] \\ &= \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) = 1] \\ &+ \frac{1}{2} * Pr[A^{\mathsf{hy}}(pk, \mathsf{Encaps}^{(1)}_{pk}(1^n, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) = 0]\end{aligned} \tag{12}$$

Summing the equations (10) , (11), (12) and using the fact that the sum of three negligible functions is negligible , we see that there exists a negligible function negl such that

$$\begin{aligned}\frac{3}{2} + \mathsf{negl}(n) \geq \frac{1}{2} * (&Pr[A^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0)) = 0] + Pr[A^{\mathsf{hy}}, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0)) = 1] \\ &+ Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0)) = 0] + Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) = 1] \\ &+ Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0)) = 1] + Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1)) = 0])\end{aligned} \tag{13}$$

where c = $\mathsf{Encaps}^{(1)}_{pk}(1^n$ in all the above . Note that

$$Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0) = 1] + Pr[A^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_0) = 0] = 1,$$

since the probabilities of complementary events always sum to 1. Similarly ,

$$Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1) = 1] + Pr[A^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k'}(m_1) = 0]) = 1,$$

Therefore ,

$$\begin{aligned}\frac{1}{2} + \mathsf{negl}(n) &\geq \frac{1}{2} * (Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k}(m_0) = 0] + Pr[\mathsf{A}^{\mathsf{hy}}(pk, c, \mathsf{Enc}^{\mathsf{SKE}}_{k}(m_1) = 1]) \\ &= Pr[\mathsf{PubK}^{\mathsf{eav}}_{\mathsf{A}^{\mathsf{hy}},\Pi^{\mathsf{hy}}}(n) = 1]\end{aligned}$$

This completes the proof of the theorem. ■

3 El Gamal like KEM

In the previous lecture we have seen the construction of El Gamal Encryption scheme based on the DDH assumption. A variant of this is to use El Gamal encryption as a part of hybrid encryption scheme. For example , the sender could choose a uniform group element m $\in$ G, encrypt this using a private-key encryption scheme and use hash of that element as the key. However this is redundant since we know that in El Gamal encryption c_1^x is indisinguishable from a uniform group element , so the sender/receiver may as well use that .Such a resulting encapsulation consists of only a single group element not two as in the case of El Gamal encryption.The construction of an El Gamal like KEM is as follows :

Construction of El Gamal like KEM

- Gen : on input 1^n , run G(1^n) to obtain (G, q, g), choose a uniform x $\in$ Z_q and set h:= g^x. Also specify a function **H** : G $\rightarrow$ $\{0,1\}^m$. The public key is $\langle$ G, q, g, h, **H**$\rangle$ and the private key is $\langle$ G, q, g, x $\rangle$.
- Encaps : on input a public key pk = $\langle$ G, q, g, h, H$\rangle$ choose a uniform y $\in$ Z_q and output a ciphertext g^y and the key H(h^y) = H(g^{xy}) .
- Decaps : on input a private key sk = $\langle$ G, q, g, h, H$\rangle$ and a ciphertext c $\in$ G , output the key H(c^x) = H(g^{xy})

The El Gamal like KEM is an improvement over El Gamal because of two major advantages achieved . Firstly the ciphertext now contains only of a single element rather than two as in the case of El Gamal . Secondly the computation is much less expensive than El Gamal since now there is no need to choose m randomly and also hashing is used in El Gamal like KEM in contrast to the multiplication used earlier in El Gamal.The security of El Gamal was based on DDH assumption - We analogously define HDH assumption on which El Gamal like KEM is based on.

Definition 3 HDH(Hash Diffie-Hellman) Assumption

HDH problem is hard relative to (G, o) and hash function **H**: G $\rightarrow$ $\{0,1\}^m$ if for every PPT A (it is hard to distinguish **H**(g^{xy}) from a random string r from $\{0,1\}^m$ even given g^x, g^y.

$$|Pr[A(G, o, q, g, g^x, g^y, \mathbf{H}(g^{xy})) = 1] - Pr[A(G, o, q, g, g^x, g^y, r) = 1]| \leq \mathsf{negl}().$$

HDH assumption is that there exists a group and a hash function **H** so that HDH is hard relative to them. $\diamondsuit$

HDH is weaker than DDH but stronger than CDH when hash function is implemented using known practical hash functions. Also, if the HDH assumption holds , then El Gamal like KEM is CPA secure . The proof is a straightforward reduction and similar to the CPA security proof of El Gamal based on DDH as seen in the previous lecture.

4 CCA security in public-key setting

In the Chosen-Ciphertext attack , the adversary is able to obtain decryption of arbitrary ciphertexts of its choice. These are more significant in the public key setting rather than private key setting . In the private symmetric key setting , a receiver intends to communicate only with a single known sender and the message encrypted with the secret key can only originate from this sender with whom the secret key was shared. However in the public key setting , a receiver might receive encrypted messaged from multiple sources unknown in advance since they have access to the public key.Thus launching CCA attacks in public key world is easier and CCA security of PKE s is a matter of high importance.

4.1 Security of PKE s against Chosen-Ciphertext Attacks

We define CCA security for public key setting analogous to the definition from private key setting . Given a public key encryption scheme π and an adversary A , we define the public key CCA indistinguishability experiment below.

The CCA indistinguishability experiment $\mathsf{PubK}^{cca}_{A,\Pi}(n)$ **:**

- $\mathsf{Gen}(1^n)$ is run to obtain keys (pk,sk)
- The adversary A is given pk and access to decryption oracle $\mathsf{Dec}_{sk}(.)$. It outputs a pair of messages m_0 and m_1 of the same length.
- A uniform bit b $\in$ {0,1} is chosen, and then a ciphertext c $\leftarrow \mathsf{Enc}_{pk}(m_b)$ is computed and given to A.
- A continues to interact with the decryption oracle, but may not request a decryption of c itself. Finally , A outputs a bit b'.
- The output of the experiment is defined to be 1 if b' = b , and 0 otherwise.

Definition 4 A public-key encryption scheme $\pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has indistiguishable encryptions under a chosen-ciphertext attack (or is CCA secure) if for all probabilistic polynomial time adversaries A there exists a negligible function negl such that

$$Pr[\mathsf{PubK}^{cca}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

$\diamondsuit$

4.2 An issue related to CCA attacks : Non - malleability

An issue that is closely related to CCA security is potential *malleability* of ciphertexts.

Definition 5 An encryption scheme (symmetric/asymmetric) is malleable if it has the following property - Given an encryption c of some unknown message m , it is possible to

come up with a ciphertext c' that is an encryption of a message m' that is related in some known way to m. ♢

For example, perhaps given an encryption of m, it is possible to construct an encryption of 2m. It is clear from the definition of malleability that a scheme is CCA secure iff it is non malleable . This follows from the fact that if the scheme is malleable , the adversary in the CCA game on receiving the challenge ciphertext $c^* \leftarrow \mathsf{Enc}(m_b)$ can query the decryption oracle on c' $\leftarrow \mathsf{Enc}(f(m_b))$ and obtain f(m_b). A real-life scenario of the use of malleability to launch an attack is as follows : Consider an e-auction among two bidders A and B who submit their bids by encrypting using a public key of R who is running the auction. If a malleable encryption scheme is used , it may be possible for the bidder A to always place the highest bid (without bidding the maximum) by carrying out the following attack - A waits till B submits the ciphertext c corresponding to his/her bid m (unknown to A) . Then A will now send a cipherext c' corresponding to the bid m' = 2m. m is still unknown to A but using the malleability property of the scheme , A is guaranteed to win. CCA secure schemes are not vulnerable to such attacks. The El Gamal scheme that we have seen is malleable and thus not CCA secure. This is because , once we are given an El Gamal encryption (c_1,c_2) of m under the public key h , the adversary can easily come up with the ciphertext c' corresponding to 2m which is nothing but (c_1,$2c_2$)

4.3 CCA Multi-message security

We define CCA Multi-message security for public key setting similar to single message CCA security. The only difference is that a vector of messages are communicated rather than a single message. Given a public key encryption scheme π and an adversary A , we define the public key CCA-multiple indistinguishability experiment.

The CCA indistinguishability experiment $\mathsf{PubK}^{\mathsf{cca}-mult}_{\mathsf{A},\Pi}(n)$ **:**

- $\mathsf{Gen}(1^n)$ is run to obtain keys (pk,sk)
- The adversary A is given pk and access to decryption oracle $\mathsf{Dec}_{sk}(.)$ which returns the plaintext message vector corresponding to the ciphertext vector queried. Then the adversary outputs a pair of message vectors $m_{0,t}$ and $m_{1,t}$ of the same length where t is the number of messages in a vector.
- A uniform bit b $\in$ {0,1} is chosen, and then a ciphertext c $\leftarrow \mathsf{Enc}_{pk}(m_{b,t})$ is computed and given to A.
- A continues to interact with the decryption oracle, but may not request a decryption of c itself. Finally , A outputs a bit b'.
- The output of the experiment is defined to be 1 if b' = b , and 0 otherwise.

Definition 6 A public-key encryption scheme $\pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has indistiguishable multiple encryptions under a chosen-ciphertext attack (or is CCA-mult secure) if for all

probabilistic polynomial time adversaries A there exists a negligible function negl such that

$$Pr[\mathsf{PubK}^{\mathsf{cca}-mult}_{\mathsf{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

♢

Theorem 2 *If a public-key encryption scheme π is CCA-secure, then it also has indistinguishable multiple encryptions under a chosen-ciphertext attack.*

The proof uses hybrid argument and is similar to the one seen in the previous lecture for the analogous theorem for CPA security.

We now know that single-message CCA security implies multi-message CCA security. Let us now look at how to construct a CCA secure PKE for long message given a CCA secure scheme π for a single bit/small message.

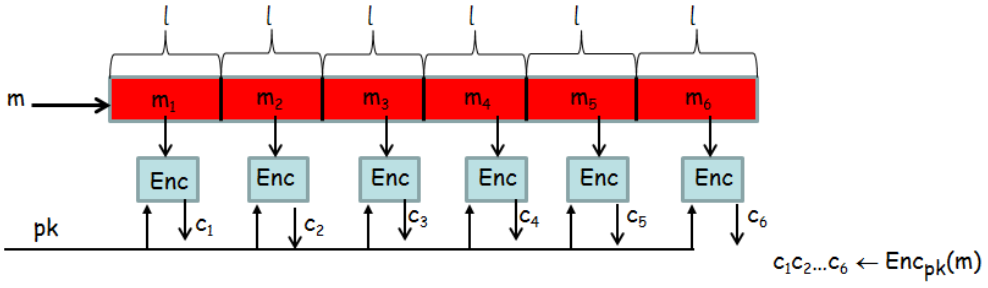

Figure 4: Trivial construction of PKE for long message

Consider the trivial construction π' as shown in which the long message is divided into blocks of fixed length and the ciphertext corresponding to each block is computed using π. This scheme is not CCA secure since the adversary can take the decryption oracle service and get the message corresponding to the ciphertext. Now truncation will give a valid ciphertext which can be used by the adversary to win the CCA game. Thus constructing CCA secure PKE for long message is not trivial and this construction can be seen in the following paper - *Steven Myers, Abhi Shelat: Bit Encryption Is Complete. FOCS 2009: 607-616*

4.4 CCA security of Hybrid Encryption using KEM

We have seen that the Hybrid π^{Hyb} of a CPA-secure KEM π and a COA-secure SKE π^{SKE} is a CPA-secure PKE. Let us look at the conditions needed for CCA security of the hybrid scheme . Suppose π^{SKE} is malleable like in the case of PRG/PRF scheme . Then the ciphertext output by the hybrid scheme is say of the form (c , G(k) $\oplus$ m) where c is the KEM ciphertext and G(k) $\oplus$ m is the SKE ciphertext where G is the PRF/PRG used in private key encryption scheme. Clearly the hybrid scheme is also malleable since the adversary can come up with a valid ciphertext easily . Thus π^{SKE} should be CCA secure. Also , π should be CCA secure as well for the hybrid scheme to be CCA secure.We state the following theorem without proof

Theorem 3 *If π is CCA-secure KEM and π^{SKE} is a CCA-secure PKE , then π^{Hyb} is a CCA secure public-key encryption scheme.*

This theorem only highlights the sufficient conditions for a CCA hybrid encryption. Please note that these conditions may not be necessary for a hybrid encryption to be CCA secure. It is known that CCA secure SKE is a must for the hybrid scheme to be secure , but weaker than CCA secure KEM may also result in a CCA secure hybrid encryption scheme.