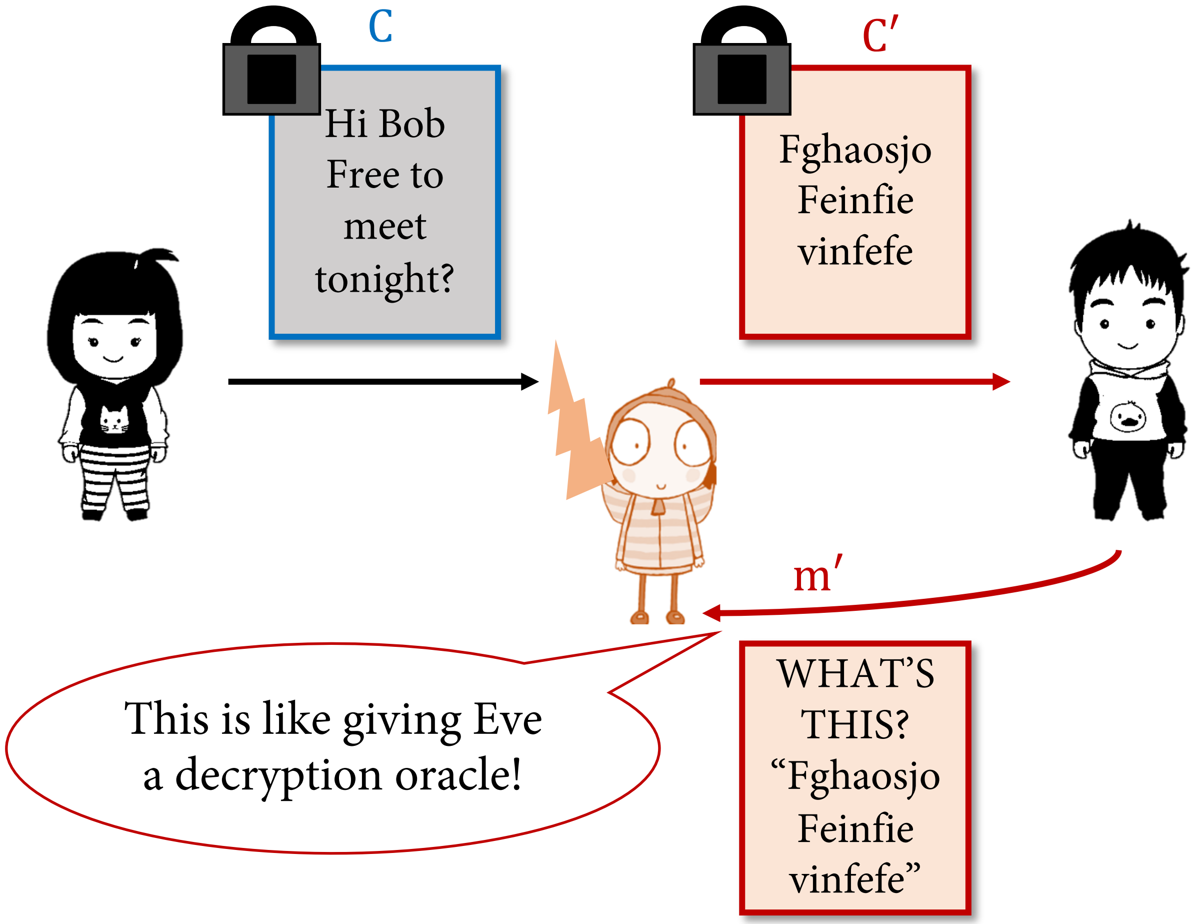
PUBLIC KEY ENCRYPTION

Lecture II

ACM Summer School 2024

Chosen Ciphertext Attack

- Suppose Alice sends an IND-CPA secure encryption of email m, i.e., the ciphertext C, to Bob.
- Eve can modify the ciphertext to C' but doesn't know what the modified email m' is.
- Bob sends back an email to ask what the message m' means?



Eve can use the related m' to potentially learn the message m!

Malleability Attacks

What if Eve "malleates" C to produce a new ciphertext C', that would decrypt to a "related" message m'?

Malleability of ElGamal Encryption

KeyGen: Uses Gen to get $(\mathbb{G}, q, g)$, $x \leftarrow \mathbb{Z}_q$

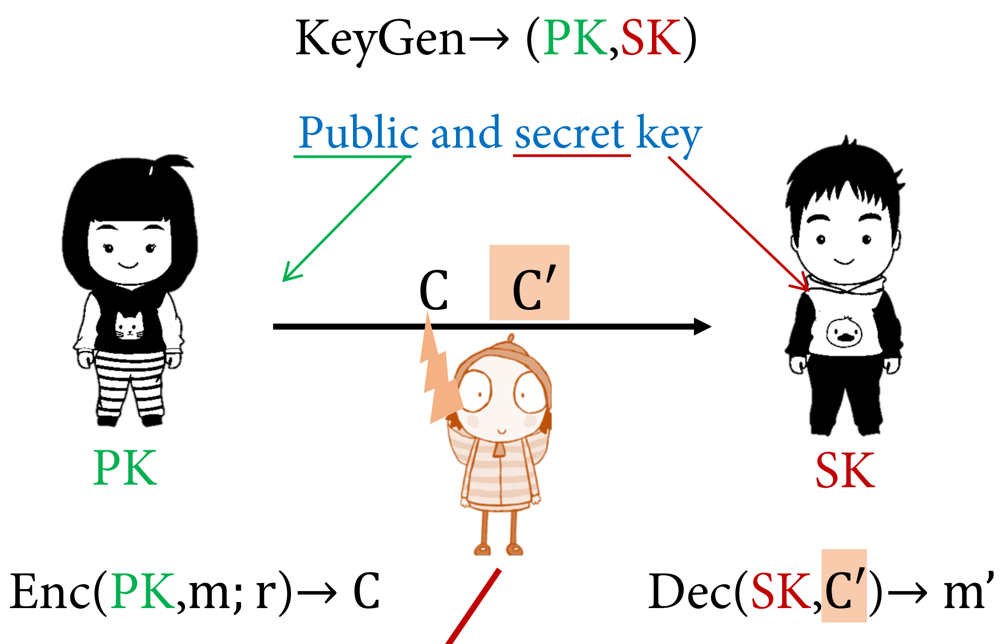
$PK = (\mathbb{G}, g, X = g^x)$, $SK = (\mathbb{G}, g, x)$

Enc(PK, m): $y \leftarrow \mathbb{Z}_q$

$C = (Y = g^y, mX^y)$

$C' = (Y = g^y, TmX^y)$

Dec(SK, C') = Tm



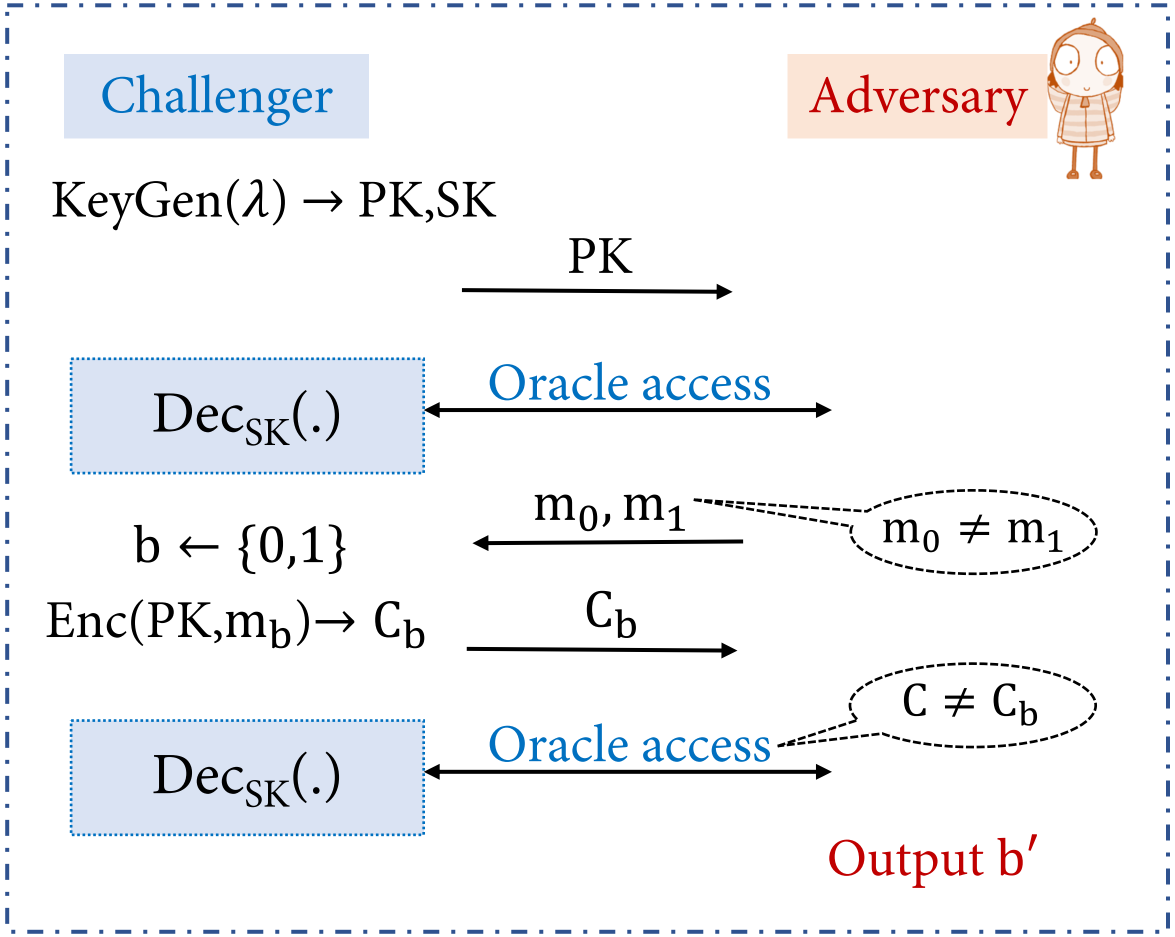
With chosen-ciphertext attack, Eve can learn Tm and hence can learn m!

IND-CCA Security for PKE

- Recall IND-CPA game only had encryption access through PK.
- For the IND-CCA game, Eve also gets decryption oracle access.
- Eve cannot query the decryption oracle for the challenge ciphertext C_b

IND-CCA Security (Chosen Ciphertext Attack)

$\forall$ PPT adversaries, $\Pr[b' = b] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$

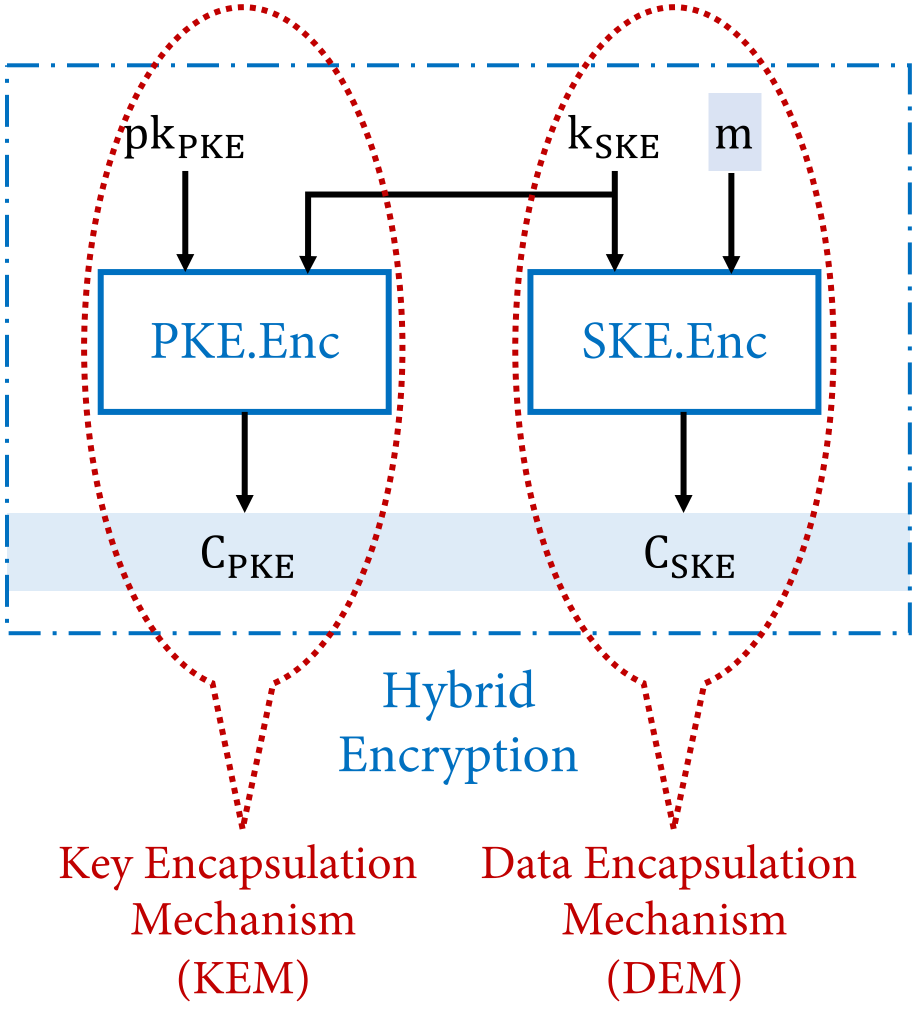


Hybrid Encryption

- PKE is far less efficient that SKE.
  E.g. DDH-based and RSA CCA encryptions all use exponentiations in group, etc.
- SKE and MAC (e.g. using block ciphers like AES) are very fast.

- Hybrid Encryption: Use CCA PKE to transfer k_{SKE} for CCA SKE. The CCA SKE is used to encrypt the message m.

- Why is this cost saving?
  PKE only used to encrypt short k_{SKE}!



Hybrid Encryption: KEM/DEM Paradigm

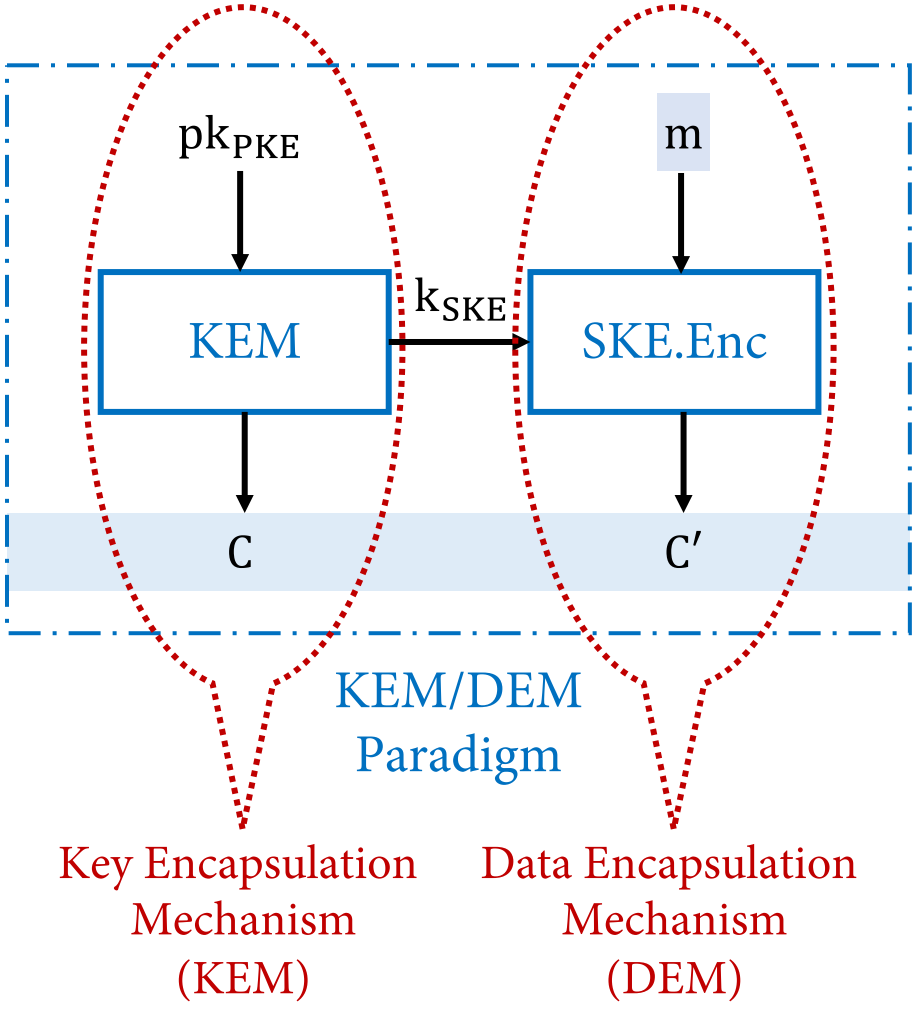
- Key Encapsulation Method (KEM): The Encapsulation process takes only the public key pk_{PKE} (no message) and directly outputs the ciphertext C and a key k_{SKE}.

- Data Encapsulation Method (DEM): The symmetric encryption used to encrypt the data/message.

For what KEM/DEM is a hybrid encryption CCA Secure?

CCA KEM + CCA SKE $\Longrightarrow$ CCA PKE

EXERCISE 6



IND-CCA Security for KEM

IND-CCA Security (Chosen Ciphertext Attack)

$\forall$ PPT adversaries, $\Pr[b' = b] \leq \frac{1}{2} + \text{negl}(\lambda)$

For what KEM/DEM is a hybrid encryption CCA Secure?

CCA KEM + CCA SKE $\Longrightarrow$ CCA PKE

EXERCISE 6

Challenger

Adversary

$\text{KeyGen}(\lambda) \to \text{PK},\text{SK}$

$\xrightarrow{\text{PK}}$

$\text{Decap}_{\text{SK}}(.)$ $\xleftrightarrow{\text{Oracle access}}$

$k_0, C \leftarrow \text{Encap}(\text{PK})$

$k_1 \leftarrow K$

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

$\xrightarrow{k_b, C}$

$\text{Decap}_{\text{SK}}(.)$ $\xleftrightarrow{\text{Oracle access}}$

$Ct \neq C$

Output b'

EXAMPLE OF AN IND-CCA SECURE PKE

Random Oracle Model [2002]

Random Oracle is a mythical public oracle RO that implements a (truly) random function:

- Public: Access to same RO is public (adversary can also query). Anyone can query x and get RO(x) in response.
- Queries are private: If a honest party queries RO(x), then the adversary does not know x!
- Implements a truly random function: On fresh query x, RO picks a random $y \leftarrow \{0,1\}^n$, returns y and adds (x, y) to a list Q of queried values. For each query x, RO first checks if x belongs to Q, in which case, it returns the corresponding y.

$$x \in \{0,1\}^k \longrightarrow \boxed{\text{RO}} \longrightarrow \text{RO}(x) \in \{0,1\}^n$$

Query (private)

Access to RO (public)

If $(x, y) \in Q$, then $\text{RO}(x) = y$
Else pick $y \leftarrow \{0,1\}^n$, and $\text{RO}(x) := y$

$Q = \{(x_i, y_i) : \text{RO}(x_i) = y_i\}$

Why Random Oracle Model?

- RO is a theoretical model, introduced as an assumption to prove security of cryptographic schemes (security definitions adapted for ROM).

- What security in ROM does not guarantee?
  There are schemes (e.g. signature and encryption schemes) secure in ROM, that are insecure in the standard model (without RO), regardless of how the RO is instantiated.

- What security in ROM tells us?
  Scheme is secure against attacks that treat RO as a black-box.
  Hence, any attack on the scheme in real world represents a weakness of the instantiation of the RO, rather than a weakness of the scheme itself!

It's good to have a security in ROM (when nothing else known).
Schemes secure in ROM are much more efficient than schemes secure in the standard model.

Recall: Discrete Log (DLog) Assumption

Discrete Log (w.r.t. g): $\mathrm{DL_g(X)} \coloneqq$ unique x such that $\mathrm{X} = \mathrm{g^x}$

Cyclic Group $(\mathbb{G}, q, \mathrm{g})$

Group Order Generator

Challenger

Adversary

$\mathrm{Gen}(1^{\lambda}) \longrightarrow (\mathbb{G}, q, \mathrm{g})$

$\mathrm{X} \leftarrow \mathbb{G}$

$\xrightarrow{(\mathbb{G}, q, \mathrm{g}), \mathrm{X}}$

$\xleftarrow{\mathrm{x}}$

Output: 1, if $\mathrm{g^x} = \mathrm{X}$
0, otherwise

Discrete Log Assumption

$\forall$ PPT Adversaries, $\Pr[\mathrm{Output} = 1] \leq \mathrm{negl}(\lambda)$

Diffie-Hellman Assumptions

Recall: Decisional Diffie-Hellman (DDH) Assumption

Cyclic Group $(\mathbb{G}, q, \mathrm{g})$

Group Order Generator

$$\{(\mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{xy}})\}_{(\mathbb{G},q,\mathrm{g})\leftarrow \mathrm{Gen}(1^{\lambda}),\, \mathrm{x,y}\leftarrow \mathbb{Z}_q} \approx_c \{(\mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{r}})\}_{(\mathbb{G},q,\mathrm{g})\leftarrow \mathrm{Gen}(1^{\lambda}),\, \mathrm{x,y,r}\leftarrow \mathbb{Z}_q}$$

Computational Diffie-Hellman (CDH) Assumption

Challenger

Adversary

$\mathrm{Gen}(1^{\lambda}) \longrightarrow (\mathbb{G}, q, \mathrm{g})$

$\mathrm{x, y} \leftarrow \mathbb{Z}_q$

$(\mathbb{G}, q, \mathrm{g}), \mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}$ →

← Z

Output: 1, if $\mathrm{Z} = \mathrm{g}^{\mathrm{xy}}$
0, otherwise

$\forall$ PPT Adversaries, $\Pr[\mathrm{Output} = 1] \leq \mathrm{negl}(\lambda)$

EXERCISE 7

DDH Assumption $\Longrightarrow$ CDH Assumption

CDH $\Longrightarrow$ DDH? No! E.g.: $\mathbb{Z}_p^*$

Diffie-Hellman Assumptions

Gap-CDH Assumption

Challenger

Adversary 

$\mathrm{Gen}(1^{\lambda}) \longrightarrow (\mathbb{G}, q, \mathrm{g})$

$\mathrm{x}, \mathrm{y} \leftarrow \mathbb{Z}_q$

$(\mathbb{G}, q, \mathrm{g}), \mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}} \longrightarrow$

$\mathcal{O}(.)$ $\longleftarrow$ (U, V)

$\xrightarrow{1,\ \text{if } V = U^{y}}$
0, otherwise

Oracle access

$\longleftarrow$ Z

Output: 1, if $\mathrm{Z} = \mathrm{g}^{\mathrm{xy}}$
0, otherwise

CDH is hard, even given an oracle that solves DDH

$\forall$ PPT Adversaries, $\Pr[\mathrm{Output} = 1] \leq \mathrm{negl}(\lambda)$

Diffie-Hellman Integrated Encryption Scheme

(DHIES) IND-CCA Hybrid Encryption

KeyGen: Uses Gen to get $(\mathbb{G}, q, g)$, $x \leftarrow \mathbb{Z}_q$, $X = g^x$, specify a function $H: \mathbb{G} \rightarrow \{0,1\}^{2n}$

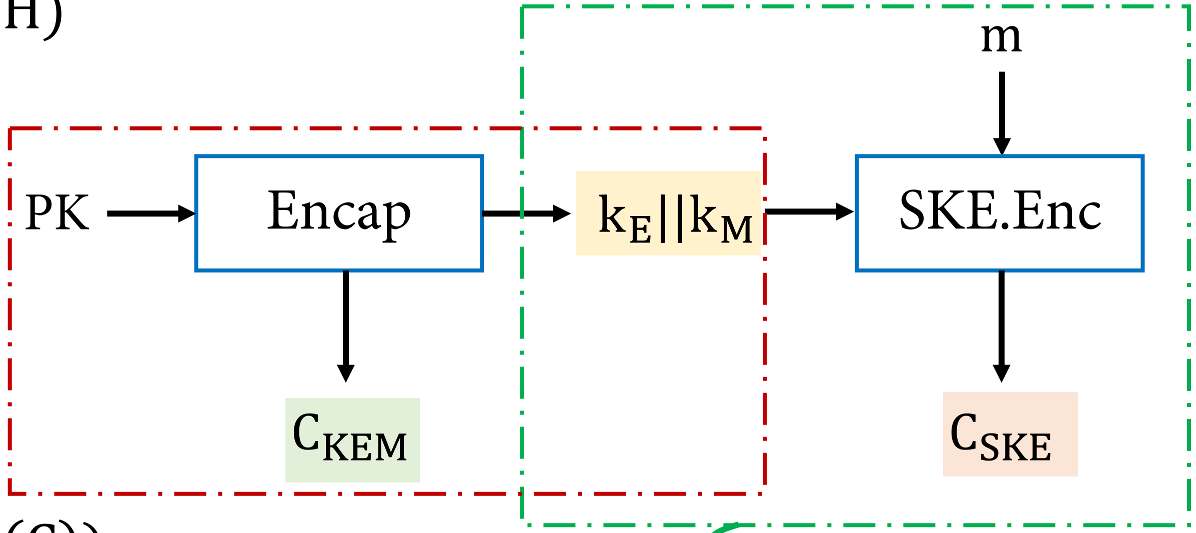
$PK = (\mathbb{G}, q, g, X, H)$, $SK = (\mathbb{G}, q, g, x, H)$

Encap(PK): $y \leftarrow \mathbb{Z}_q$

$k_E || k_M \leftarrow H(X^y)$

$C_{KEM} = g^y$

CCA KEM?



CCA Secure SKE

SKE.Enc($k_E || k_M$, m):

$C_{SKE} = (C = Enc_{k_E}(m), MAC_{k_M}(C))$

CPA Encrypt then MAC → CCA SKE

(Sikhar's talk)

<u>RECALL</u>: CCA KEM + CCA SKE ⟹ CCA PKE

DHIES: IND-CCA KEM

KeyGen: Uses Gen to get $(\mathbb{G}, q, g)$, $x \leftarrow \mathbb{Z}_q$, $X = g^x$, specify a function $H: \mathbb{G} \to \{0,1\}^{2n}$
$PK = (\mathbb{G}, q, g, X, H)$, $SK = (\mathbb{G}, q, g, x, H)$

Encap(PK): $y \leftarrow \mathbb{Z}_q$
$k \leftarrow H(X^y)$; $C_{KEM} = g^y$
Output (k, C_{KEM})

Decap(SK,C_{KEM}): $H(C_{KEM}^x)$

THEOREM:

If gap-CDH is hard for the collection of groups used and H is modeled as an RO, then the above construction is an IND-CCA secure KEM.

DHIES: IND-CPA KEM

THEOREM:

If CDH is hard for the collection of groups used and H is modeled as an RO,
then the above construction is an IND-CPA secure KEM.

PROOF SKETCH:

CDH Adversary A^* (acts as IND-CPA Challenger) — IND-CPA KEM Adversary A

- Gets challenge $(\mathbb{G}, q, g), g^x, g^y$.
- Set PK$= (\mathbb{G}, q, g), g^x$, H.
- Pick k at random, set $C = g^y$.

- A's queries: $Z_1, .., Z_t$
  Pick $i \in [t]$ and output Z_i as the CDH guess.

$\xrightarrow{\text{PK}}$

$\xrightarrow{\text{k, C}}$

Random Oracle queries $\longleftrightarrow$

If A didn't query g^{xy} then, k is uniform from A's perspective! Hence, can guess b w.p. 1/2

If A did query g^{xy} then CDH is broken w.p. 1/t!

EXERCISE 8: Formalize this proof!

DHIES: IND-CCA KEM

THEOREM:

If gap-CDH is hard for the collection of groups used and H is modeled as an RO, then the above construction is an IND-CCA secure KEM.

PROOF SKETCH:

- CCA KEM Adversary has Oracle access to $\text{Decap}_{SK}(.)$, which in turn means that it has access to a DDH solver.
- Excluding the $\text{Decap}_{SK}(.)$, the CCA KEM Adversary is like a CPA adversary. For this, we already saw that we can reduce the security to CDH.
- Thus, as long as CDH is hard, even given a DDH solver (a.k.a. gap-CDH), CCA KEM security holds for this scheme.

Other IND-CCA KEM Schemes

- Fujisaki-Okamoto
  Another Hybrid Encryption Scheme secure in ROM

- RSA-OAEP
  Secure in ROM (we didn't look at RSA assumption today)

- Cramer-Shoup Encryption
  Provably secure CCA scheme under DDH

Introduction to Modern Cryptography, Katz and Lindell
(Chapter 11)